

Aplicación de la metodología de desarrollo ágil Scrum para el desarrollo de un sistema de gestión de empresas



Universidad
Carlos III de Madrid

TRABAJO FIN DE GRADO

GRADO EN INGENIERÍA INFORMÁTICA

Madrid, Septiembre 2015

Autor: Aitor Urteaga Pecharromán.

Tutor: Javier Fernández Muñoz.

Agradecimientos

A todas las personas que me han apoyado durante esta etapa universitaria, especialmente en los momentos más difíciles, momentos en los que lo más fácil hubiese sido rendirse pero gracias a vuestra ayuda seguí adelante.

Ahora finalmente he conseguido alcanzar la meta que me propuse hace años y tengo que reconocer no lo habría logrado sin vuestro apoyo.

Muchas gracias a todos por estar ahí.

Resumen

Este proyecto describe el desarrollo de un sistema de gestión de empresas, también conocido como ERP. El cual se desarrolla empleando la metodología ágil Scrum. El proyecto se divide en varias iteraciones en las cuales se obtiene como resultado un prototipo. El sistema está compuesto por una aplicación cliente para uso del usuario y una base de datos centralizada donde se almacenarán todos los registros relativos a la empresa.

Abstract

This project describes the development of a enterprise resource planning, also known as ERP. Which is developed using agile methodology Scrum. The project is divided into several iterations in which you get as a result a prototype. The system consists of a client application to use the user and a centralized database where all records relating to the company store.

Contenido

Capítulo 1: Introducción	9
Contexto	9
Motivaciones	10
Objetivo	11
Estructura	12
Capítulo 2: Estado del arte	13
Aplicaciones de gestión empresarial.....	13
ERP	13
Ciclo de vida.....	14
Implementación Big Bang	15
Implementación Ágil	18
Análisis.....	18
Realización del Sprint	19
Productos existentes	20
Herramientas de desarrollo	23
Microsoft Windows 8.1	24
Visual Studio 2013	24
SGBD	25
Metodología de desarrollo	26
¿Qué es Scrum?	26
¿Qué caracteriza a Scrum?	27
Los papeles de Scrum	28
Capítulo 3: Análisis del sistema	31

Definición.....	32
Historias de usuario.....	33
Tareas	38
Estimación	49
Planning Poker.....	50
Estimación en el proyecto	50
Capítulo 4: Desarrollo del sistema	53
Visión general	53
Arquitectura	54
Cliente - Servidor	54
MVC	55
Capa Datos.....	58
Sprints.....	63
Sprint Planning	64
Sprint 0	65
Primer Sprint	67
Segundo Sprint	84
Tercer Sprint.....	93
Cuarto Sprint	101
Capítulo 5: Planificación y presupuesto	113
Planificación	113
Presupuesto.....	114
Resumen de tiempo dedicado	114
Coste de personal.....	115
Coste de Hardware.....	115

Coste de Software	116
Resumen de costes.....	116
Capítulo 6: Conclusiones	117
Desarrollos futuros	119
Conclusiones personales	122
Anexo A: Manual de usuario	123
Anexo B: Bibliografía	129

Ilustración 1- ERP Ciclo de vida	14
Ilustración 2 - Rendimiento del sistema.....	16
Ilustración 3 - Proceso de implantación de Big Bang.....	17
Ilustración 4 - Procesos Scrum	29
Ilustración 5 - Procesos Scrum (complejo).....	30
Ilustración 6 - Cliente-Servidor.....	54
Ilustración 7 - MVC	56
Ilustración 8 - Capas	57
Ilustración 9 - Capas desarrollo.....	70
Ilustración 10 - Base de datos	70
Ilustración 11 - Modelo base datos I.....	72
Ilustración 12 - Modelo base de datos II.....	73
Ilustración 13 - Modelo base de datos III.....	74

Ilustración 14 - Procedimiento buscar	75
Ilustración 15 - Procedimiento insertar	75
Ilustración 16 - Procedimiento eliminar.....	76
Ilustración 17 - Procedimiento mostrar	76
Ilustración 18 - Procedimiento editar	76
Ilustración 19 - Conexión base de datos	77
Ilustración 20 - Interfaz de usuario Grupo de artículos	80
Ilustración 21 - Interfaz de usuario Artículos	81
Ilustración 22 - Interfaz de usuario Proveedores.....	87
Ilustración 23 - Interfaz de usuario Centros.....	89
Ilustración 24 - Interfaz de usuario Stocks.....	90
Ilustración 25 - Interfaz de usuario Ventas	96
Ilustración 26 - Interfaz de usuario Empleados	97
Ilustración 27 - Interfaz de usuario Clientes	98
Ilustración 28 - Interfaz de usuario Compras.....	104
Ilustración 29 - Interfaz de usuario Inicio.....	105
Ilustración 30 - Interfaz de usuario Venta rápida	106
Ilustración 31 - Interfaz de usuario Compra rápida	106
Ilustración 32 - MS Excel Obj. Library.....	107
Ilustración 33 - Interfaz de usuario Importar	108
Ilustración 34 - Diagrama de Gantt	113
Ilustración 35 - Comparativa esfuerzos.....	118
Ilustración 36 - Partes aplicación	120

Capítulo 1: Introducción

El proyecto, en el cual se basa este documento, consiste en la utilización de la metodología ágil Scrum en el desarrollo de un sistema de gestión de pequeñas empresas del tipo ERP. En este documento se describen las distintas fases del proyecto así como el contexto que lo rodea en cuanto a tecnologías y metodologías.

Contexto

Este proyecto se sitúa dentro del área de los sistemas de información si se tiene en cuenta la siguiente definición:

“Un sistema de información es un conjunto de elementos que interactúan entre sí con el fin de apoyar las actividades de una empresa o negocio.”

En este caso se trata de un sistema de gestión interna del negocio, principalmente orientado al área logística pero fácilmente adaptable y ampliable a otros campos de la organización, como gestión de personal, tesorería, relación con los proveedores o clientes, etc.

De este modo este tipo de software se hace imprescindible en cualquier empresa a día de hoy en mayor o menor medida. La informatización de los datos y procesos relacionados al negocio o a la propia gestión interna es la tendencia en un mundo donde el acceso a un ordenador y a Internet cada vez es más generalizado o completamente implantado en la mayoría de países desarrollados.

Por otra parte para el desarrollo de la aplicación se ha optado por seguir metodologías ágiles de desarrollo de software, especialmente Scrum. Actualmente las metodologías ágiles son una tendencia en auge frente a métodos más tradicionales.

Todas las metodologías que se consideran ágiles cumplen con el manifiesto ágil que no es más que una serie de principios que se agrupan en 4 valores:

1. Los individuos y su interacción, por encima de los procesos y las herramientas.
2. El software que funciona, frente a la documentación exhaustiva.
3. La colaboración con el cliente, por encima de la negociación contractual.
4. La respuesta al cambio, por encima del seguimiento de un plan.

Con estas premisas el objetivo es incrementar la eficiencia de las personas involucradas en el proyecto, minimizando el coste de éste.

Especialmente se pueden encontrar beneficios en las metodologías ágiles frente a las clásicas en proyectos nuevos y desconocidos, es decir, proyectos en los cuales se carece de la experiencia de desarrollos similares o iguales, lo que implica innovar y adentrarse en lo desconocido. Y en este punto es donde la flexibilidad que aportan las metodologías ágiles junto a la participación del cliente a la hora de decidir los cauces que toma el desarrollo proporciona un marco muy ventajoso al proyecto, facilitando el éxito y moderando los costes.

Motivaciones

La motivación que ha impulsado la elección y desarrollo de este proyecto ha sido las ganas de aplicar diferentes conocimientos adquiridos durante el transcurso de la carrera en las diferentes asignaturas cursadas, especialmente las asignaturas propias de la especialidad elegida: “Sistemas de Información”. Además se tuvo en cuenta la experiencia profesional en el campo de los ERPs para aprovechar el conocimiento de estos sistemas desde dentro.

Por otra parte el auge del uso de metodologías ágiles en el desarrollo de software gracias a las ventajas que proporcionan frente a métodos clásicos es la causa de la elección de Scrum como referente en la forma que se ha desarrollado el proyecto.

Objetivo

El objetivo que persigue este proyecto es el desarrollo de una aplicación de escritorio que se encargue de la gestión logística y económica para profesionales autónomos o pequeñas empresas.

Además se aprovechará la oportunidad para poner en práctica la metodología Scrum, basándose principalmente en este modelo para el desarrollo de la aplicación.

El sistema a desarrollar buscará facilitar el tratamiento y almacenamiento de los datos relevantes del negocio, permitiendo de forma sencilla la introducción de los datos en el sistema y la elaboración de informes utilizando la información disponible.

De esta manera se optimizarán los procesos empresariales permitiendo el acceso a toda la información de forma precisa, confiable y asegurando la integridad de los datos. Además posibilita la oportunidad de compartir la información entre los miembros de la organización de diferentes departamentos, sedes, etc.

Este objetivo se subdivide en una serie de subobjetivos que son:

- **Utilización de una metodología ágil de desarrollo:** se busca aprovechar las ventajas que ofrecen este tipo de metodologías en el desarrollo, aprovechando su flexibilidad.
- **Desarrollo de una aplicación utilizable por el usuario medio:** con este objetivo se busca que cualquier empleado de la empresa pueda hacer uso del sistema sin necesidad de ser un usuario experto.
- **Modularidad:** la aplicación en si será una sola, pero debe tener sus funciones moduladas para facilitar tanto su mantenimiento como cualquier posible evolutivo.
- **Persistencia de la información:** los datos introducidos en el sistema deben ser persistentes entre sesiones y usuarios.

Estructura

El documento se divide en una serie de capítulos que a su vez se subdividen en apartados. Los capítulos son los siguientes:

1. **Introducción:** da una visión global del proyecto, exponiendo los objetivos a alcanzar, el porque de la realización de este proyecto y el contexto.
2. **Estado del arte:** expone el contexto tecnológico que rodea al proyecto, detallando diferentes opciones disponibles e introduciendo al lector a las principales ofertas que existen en el mercado en relación al proyecto.
3. **Análisis del sistema:** es un estudio previo al desarrollo en el cual se detalla las propiedades del sistema y los elementos que se deben desarrollar.
4. **Desarrollo del sistema:** en este capítulo se detalla todo lo relativo al propio desarrollo del proyecto.
5. **Planificación y presupuesto:** se expone la planificación previa basada en las estimaciones para la duración del proyecto y se realiza un pequeño análisis económico de éste para hallar el coste final de su realización.
6. **Conclusiones:** último capítulo del documento en el cual se plasman las conclusiones extraídas después de la finalización del proyecto, tanto personales como de resultados. También se proponen desarrollos futuros que enriquezcan al sistema.

Capítulo 2: Estado del arte

En este apartado se presentarán las tecnologías y metodologías relacionadas con el proyecto de este documento. El fin de ello es acercar al lector a estos conceptos y al contexto que rodea al proyecto en sí. De forma que pueda comprender con más facilidad el resto del documento.

El proyecto está basado en el desarrollo de un pequeño sistema de gestión de empresas empleado metodologías ágiles, y por ello en este capítulo se hablará de lo que es un sistema de planificación de recursos empresariales (ERP, por sus siglas en inglés, enterprise resource planning). Y en que consiste la metodología ágil Scrum.

Aplicaciones de gestión empresarial

ERP

Los sistemas ERP son sistemas grandes y modulares de información que engloban los diferentes procesos de las empresas, como ventas, producción, logística, y finanzas. ERP tiene dos importantes características: facilitan una conexión causal entre el modelo de visión de los procesos del negocio y la implementación de los mismos, y en segundo lugar, aseguran un nivel de integración, integridad de datos y seguridad, lo que no es muy sencillo de conseguir con múltiples plataformas software. Las dos primeras etapas del ciclo de vida de ERP son la planificación y selección del paquete. Una vez que se han completado estas fases, comienza la etapa de implementación y normalmente tiene una duración mayor que las otras dos fases anteriores juntas. Esta etapa puede durar meses o incluso años, por lo que es bastante importante y el éxito de ERP viene determinado por esta etapa y la elección de cómo abordar la implementación. Hay una variación en los rangos de fracaso para la implementación de ERP, que subyace en la propia definición de fracaso. Donde un autor define fracaso cuando se ha terminado el proyecto completo, otro autor puede definirlo cuando no se han cumplido los objetivos por tiempo y/o presupuesto.

Hay diferentes metodologías de implementación pero las dos aproximaciones más usuales son:

- Implementación Big Bang.
- Implementación ágil.

Tal como se indica en el inicio de este documento, para este desarrollo se empleará la metodología Scrum que más adelante se detallará su funcionamiento.

No obstante, en este mismo capítulo se explicará brevemente estas dos implementaciones.

Ciclo de vida

EL ciclo de vida de un ERP consiste en 4 etapas como se puede observar en la Figura 1. En ella se puede observar las dos primeras etapas (planificación y selección del paquete) y las dos últimas etapas, que son implementación y mantenimiento. La fase de planificación es en la que los gerentes del proyecto se deben cuestionar la necesidad para un nuevo sistema ERP, a la vez que seleccionan la aproximación general del sistema de información que mejor se adecuará a los desafíos críticos de la empresa y que mejorará la estrategia organizativa. Esta fase de decisión incluye la definición de los requisitos del sistema, los objetivos y beneficios. La fase de la selección del paquete consiste en la selección del producto que mejor se adecúe a los requisitos de la organización. Se analizan factores como el precio, la experiencia y los servicios de mantenimiento y se define el acuerdo contractual, pero también es importante realizar un análisis del retorno de la inversión del producto seleccionado. En la fase de implementación, se elige la estrategia de implementación, personalización y adaptación del paquete ERP adquirido de acuerdo a las necesidades de la organización. Una vez que se ha implementado el sistema, se debe hacer una labor de mantenimiento porque se deben corregir los fallos y realizar mejoras generales a los sistemas.

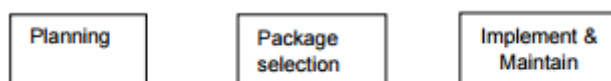


Fig. 1 ERP life cycle

Ilustración 1 - ERP Ciclo de vida

Aproximaciones en la implementación de ERP

Es muy importante tener en cuenta que la instalación y la implementación son dos conceptos completamente distintos. La instalación es el mecanismo de cambiar de un paquete software a otro, teniendo siempre la mínima cantidad de errores. La implementación es un método que usa una empresa para lograr sus objetivos cambiando la forma en la que llevan a cabo sus operaciones.

La implementación garantiza no solo que se ha instalado el software, sino que también cumple el objetivo operacional y estratégico. Una instalación con éxito no asegura que el sistema ERP logre las metas de la empresa durante un periodo de tiempo prolongado.

El éxito de ERP reside en la elección de la implementación y también en los costes y riesgos de la implementación del proyecto, que es dependiente del tipo de implementación elegido por la organización. Las dos aproximaciones de implementación más usuales son Big Bang y la ágil, como se comentó anteriormente. En la Big Bang, se divide el proceso de implementación en dos fases distintas que se ejecutan en un orden estricto, sin la posibilidad de volver a una fase anterior, ya que es muy caro rehacer los cambios de ERP para la empresa. Por lo tanto, la aproximación Big Bang también es conocida como el modelo en cascada. Por otro lado, la metodología ágil, es iterativa e incremental por naturaleza, centrándose en enviar el conjunto completo funcional mediante pequeños trozos a lo largo de ciclos cortos. La aproximación ágil también se conoce como aproximación por fases o incremental.

Implementación Big Bang

Con esta implementación el cambio entre usar el sistema heredado y el nuevo sistema es la llamada transición de sistema.

Todo el mundo comienza a usar el sistema nuevo en la misma fecha y a partir de ese momento no se usará de nuevo el antiguo sistema. La adopción de esta implementación es muy arriesgada porque hay muy pocas posibilidades de incorporar aprendizaje de este nuevo sistema. Una vez que la gerencia decide utilizar el método Big Bang y apoya los cambios necesarios para implementarlo, comienza el verdadero proceso de cambio. En este proceso existen varios pasos: preparación de la gerencia, conversión del sistema, lanzamiento de partes del sistema ERP y facilitar la enseñanza necesaria para los futuros usuarios. Ya

realizados los pasos anteriores, se apagará el sistema antiguo, y entonces, se podrá cargar y lanzar el sistema ERP [4].

Una vez que se lanza el ERP nuevo ya no hay vuelta atrás. Después de introducir el ERP, normalmente hay caída inicial del sistema, que ocurre porque los usuarios están peleándose con el nuevo ERP y a su vez con la conexión entre el sistema y su trabajo cotidiano. Lo que se puede observar en la Figura 2, es que después de la línea roja, que representa el big bang, el rendimiento organizativo tiene un valor inferior, que permanecerá un tiempo ahí, y después se volverá a incrementar su valor.

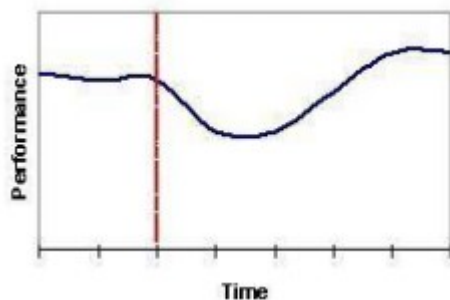


Ilustración 2 - Rendimiento del sistema

Se tiene que considerar este periodo de alcance, de forma que seamos capaces de tratar con estos problemas y dificultades. El proceso total de la implementación para los ERP de tipo big bang se muestra en la Figura 3.

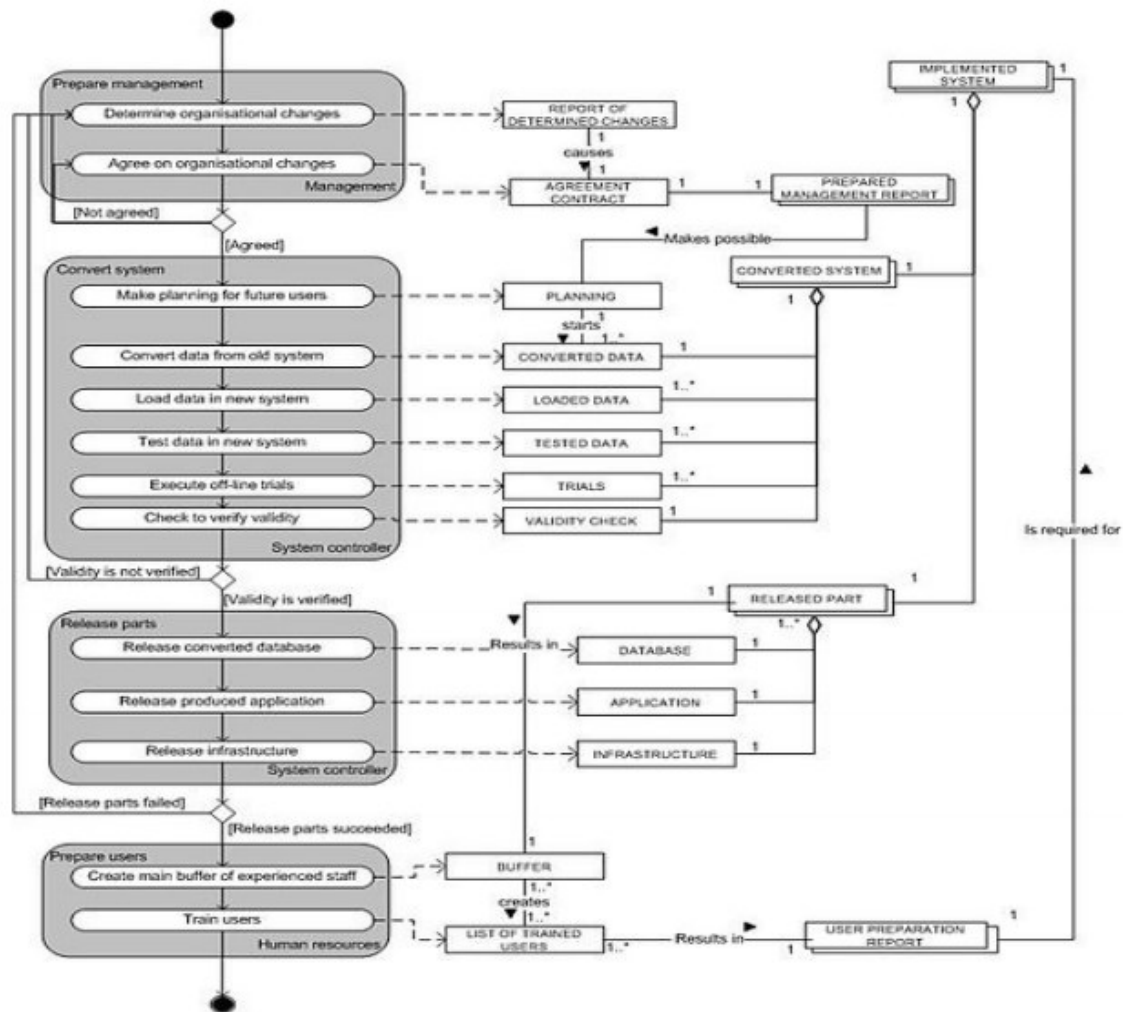
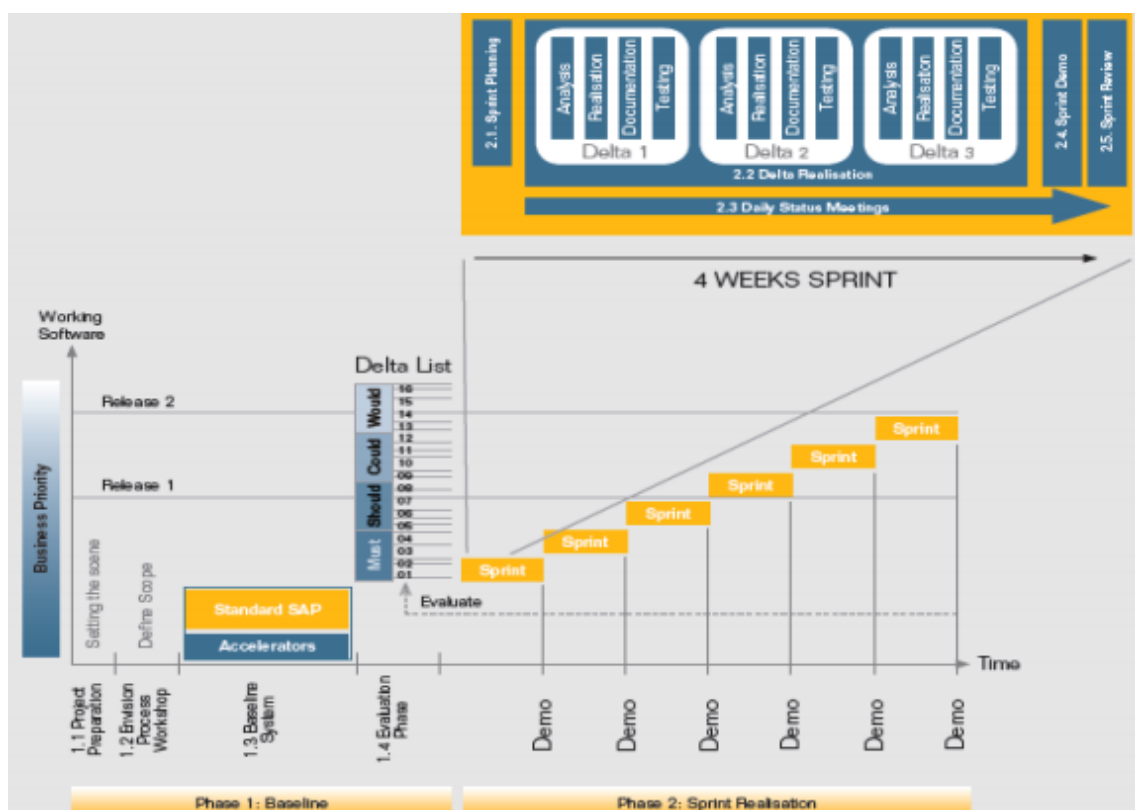


Ilustración 3 - Proceso de implantación de Big Bang

Este modelo consta de partes separadas, partiendo de la preparación de la gestión y las actividades involucradas (las cajas grises de la figura), al paso que se tiene que dar una vez que se ha llegado a un acuerdo. Este último paso consiste en la conversión del sistema con todas las subactividades incluidas y los resultados de las mismas, representados justo a la derecha de las cajas grises de la imagen. El informe de la gestión del sistema permite una mejor organización. Una vez que se ha convertido el sistema, se puede comenzar a lanzar partes del sistema (tercera caja gris de la figura), y después el siguiente paso (la última caja gris) es la preparación del usuario, que lo lleva a cabo el departamento de recursos humanos. Como resultado se obtendrá una lista de usuario expertos en el nuevo sistema. Mientras se enseña a los usuarios, se realiza un informe con esta instrucción de los usuarios, la cual se necesitará para la implementación final del sistema ERP.

Implementación Ágil

La implementación ágil se realiza de una forma incremental o por fases. El principal compromiso de la implementación de la metodología ágil es evitar los obstáculos y problemas asociados a la implementación big bang. La metodología ágil se basa en la simplicidad, en entregar las funcionalidades operativas del software lo más rápido posible, empezando por los componentes más importantes del negocio [1]. La implementación ágil consiste en dos fases: análisis y realización de Sprint como se muestra en la Figura 4.



A continuación se van a explicar ambas fases detenidamente. [2]

Análisis

Esta fase consiste en 4 pasos:

1. Preparación del proyecto durante el cual se definen los diferentes elementos, tales como labores,

responsabilidades, estándares para la documentación y requisitos del hardware.

2. Proceso de visualización en el que se identifican cuidadosamente todos los procesos operativos y procesos que dependen de condiciones específicas, como seguridad, autorizaciones e interfaces. Estos resultados serán la base para la construcción de unos cimientos sólidos para el proyecto entero.
3. Funcionamiento de referencia del sistema, el cual se basará en el software de ERP.
4. Fase de evaluación. En esta fase, se determina la prioridad de los requisitos adicionales y las funcionalidades, en orden del valor del negocio. Después de esto, el equipo de implementación estima el esfuerzo que será necesario para realizar esto y determina la planificación de los sprints para que se suministren los componentes del sistema.

Realización del Sprint

Esta fase consiste en los 5 siguientes pasos [3]:

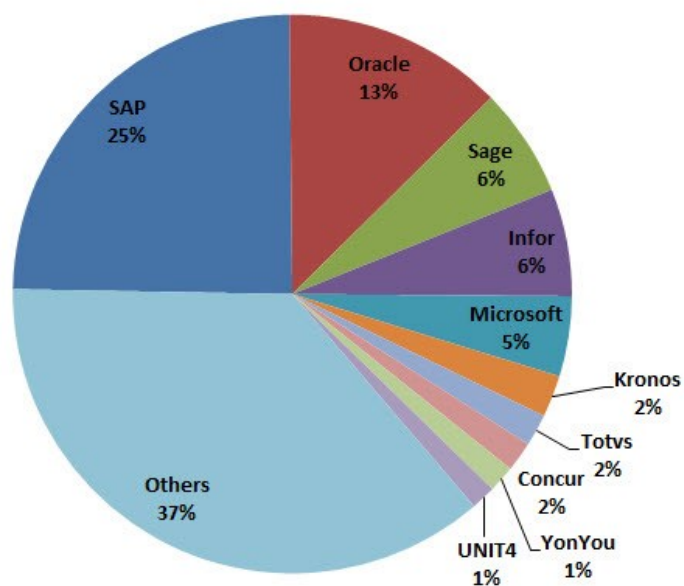
1. Reuniones de planificación al comienzo de cada sprint. Se define el objetivo del sprint entre el propietario y el equipo de implementación.
2. Realización de los requisitos, de pruebas y documentación.
3. Reuniones diarias del estado del proyecto. En esta fase se registra el estado del proyecto y se discuten los diversos obstáculos que el equipo ha podido encontrar.
4. Sesión de prueba del Sprint. Durante esta fase los usuarios y el equipo IT determinan si los procesos cumplen los requisitos.

5. Se llevará a cabo una revisión del Sprint para comprobar que se puede mejorar en los futuros sprints.

Productos existentes

Actualmente en el mercado existen muchas opciones de ERP, desde grandes sistemas para grandes clientes, hasta software libre destinado a pequeñas empresas, pasando por aplicaciones desarrolladas a medida del cliente. Entre todas estas alternativas se pueden destacar los siguientes productos por su relevancia [5]:

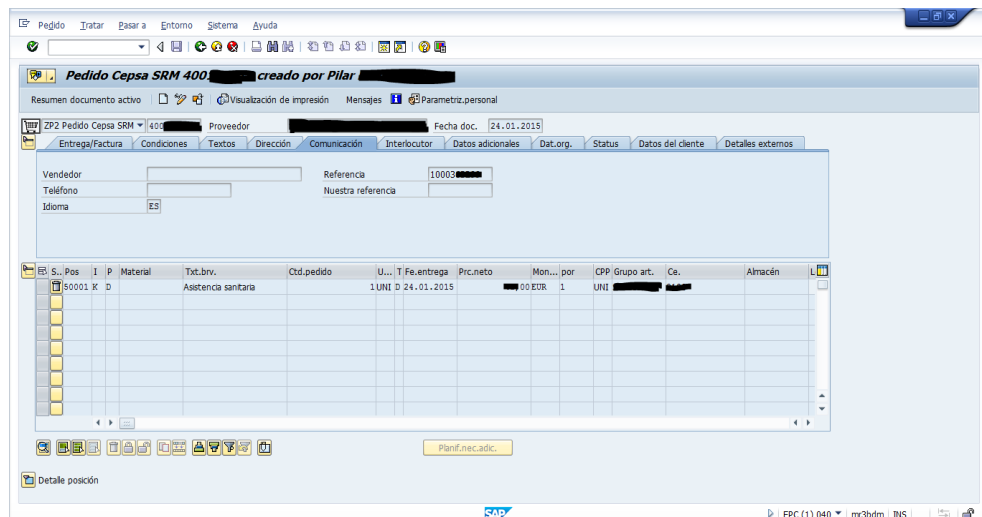
Worldwide ERP Software Market Share, 2012
Market Size: \$24.5B; 2.2% Growth Over 2011



- SAP R/3 o All in One:

Es actualmente el sistema de mayor éxito del mercado, es la solución integrada de la empresa SAP. Se encuentra dividida por módulos según su área de negocio y pudiéndose implantar por separado. Además de los módulos que forman el estándar del sistema, ofrece enormes posibilidades de desarrollos propios para hacer la solución más adaptada a los requerimientos de cada cliente. Se trata de un sistema integrado, por lo cual la información que utiliza se guarda de manera centralizada para todo el sistema, permitiendo así el manejo de la información en empresas de gran tamaño [6].

- *Pros:* mayor oferta de preconfiguraciones para diferentes verticales de la industria, velocidad de implantación, escalabilidad, mayor cuota de mercado.
- *Contras:* perfil de implementación de mayor riesgo, tiene el coste más elevado, mayor tiempo de amortización.



- Oracle ERP:

Oracle ofrece una gran cantidad de opciones de ERP. E-Business Suite 12.1 abarca todas las facetas de la ERP y todas las industrias. La última versión incluye una cartera integrada de herramientas de inteligencia empresarial. También ofrece suites completas de ERP de las empresas adquiridas como PeopleSoft y JD Edwards [6].

- *Pros:* coste más reducido que SAP, segunda mayor cuota de mercado, mejor adaptación a la nube.

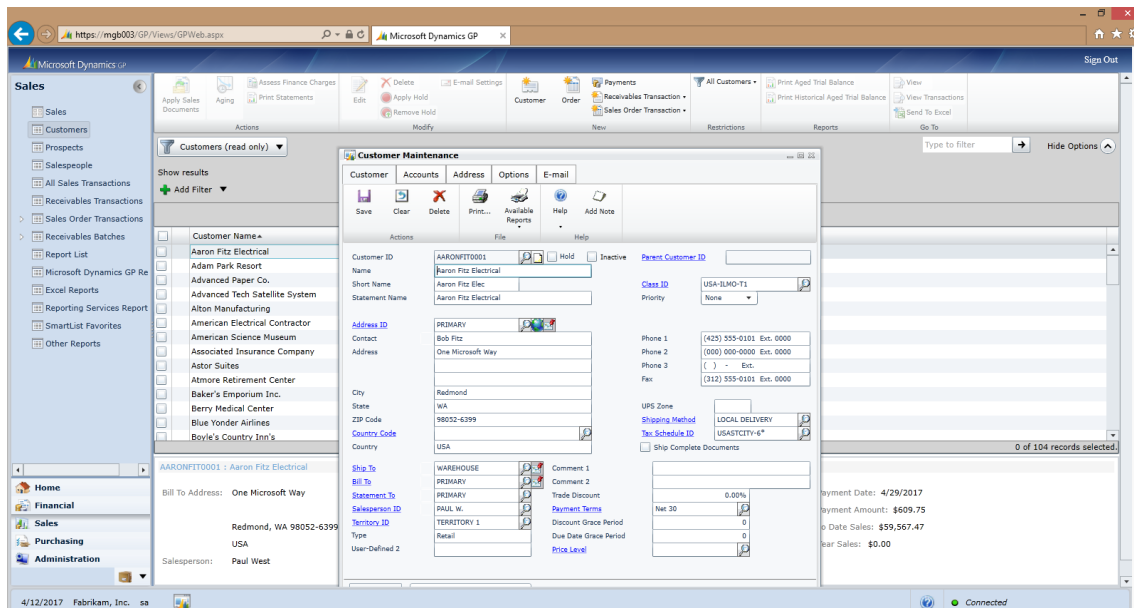
- *Contras:* menor funcionalidad, menor escalabilidad.



- Microsoft Dynamics:

Es el producto ERP y CRM de Microsoft, la rama ERP es una familia de productos de planificación de recursos empresariales dirigido a las medianas empresas así como a empresas filiales y divisiones de grandes organizaciones. Microsoft Dynamics ERP incluye 4 principales productos: Microsoft Dynamics AX (antiguamente Axapta), Microsoft Dynamics GP (antiguamente Great Plains Software), Microsoft Dynamics NAV (antiguamente Navision), Microsoft Dynamics SL (antiguamente Solomon IV) [6].

- *Pros:* mayor porcentaje de usuarios que ven reflejados sus beneficios, tiene el precio más bajo entre los tres grandes sistemas ERP del mercado. Fácil uso. Acceso online con la interfaz Microsoft Outlook.
- *Contras:* menor cuota de mercado que las alternativas, peor adaptabilidad a la empresa que sus competidores.



- Software Libre:

Además de los principales productos que se han mencionado anteriormente, también existen multitud de opciones en el ámbito del software libre, destacando estas opciones como muy interesantes para empresas y usuarios como a los que va orientado este proyecto, por eso mismo es importante destacar la existencia de estas alternativas a los grandes productos software. Algunos ejemplos de software libre son ERPNext, Apache OFBiz, Odoo, etc.

- Pros: permite ahorrar en la adquisición, mantenimiento y renovación de tecnologías, de fácil acceso para la pequeña empresa.
- Contras: menor funcionalidad, peligro de abandono por parte de la comunidad.

Herramientas de desarrollo

En este apartado se hablará de las herramientas de desarrollo relacionadas con el proyecto.

Microsoft Windows 8.1

Windows 8.1 es una versión de Microsoft Windows, línea de sistemas operativos desarrollada por Microsoft para su uso en computadoras personales, incluidas computadoras de escritorio en casa y de negocios, computadoras portátiles, netbooks, tabletas, servidores y centros multimedia.

Para el proyecto a desarrollar Windows es el entorno ideal para ello, ya que la previsión es que los equipos del cliente donde correrá la aplicación también utilicen este sistema operativo, por lo que más adelante se detallará con que tecnología se desarrollará el sistema, que será toda la suite de Microsoft, así que utilizar Windows supondrá poder utilizar todos estos programas de forma mucho más sencilla.

Visual Studio 2013

Microsoft Visual Studio es un entorno (IDE, por sus siglas en inglés) para sistemas operativos Windows. Soporta múltiples lenguajes de programación tales como C++, C#, Visual Basic .NET, F#, Java, Python, Ruby, PHP; al igual que entornos de desarrollo web como ASP.NET MVC, Django, etc., a lo cual sumarle las nuevas capacidades online bajo Windows Azure en forma del editor Monaco.

Visual Studio permite a los desarrolladores crear aplicaciones, sitios y aplicaciones web, así como servicios web en cualquier entorno que soporte la plataforma .NET (a partir de la versión .NET 2002). Así se pueden crear aplicaciones que se comuniquen entre estaciones de trabajo, páginas web, dispositivos móviles, dispositivos embebidos, consolas, etc.

En el caso del proyecto que se aquí se trata se utilizará para el desarrollo de la aplicación en el lenguaje C# y aprovechará la facilidad de uso de una base de datos usando Microsoft SQL Server 2012 como se detallará en el siguiente apartado.

Un sistema gestor de base de datos (SGBD) es un conjunto de programas que permiten el almacenamiento, modificación y extracción de la información en una base de datos, además de proporcionar herramientas para añadir, borrar, modificar y analizar los datos. Los usuarios pueden acceder a la información usando herramientas específicas de interrogación y de generación de informes, o bien mediante aplicaciones al efecto.

Estos sistemas también proporcionan métodos para mantener la integridad de los datos, para administrar el acceso de usuarios a los datos y para recuperar la información si el sistema se corrompe. Permiten presentar la información de la base de datos en variados formatos. La mayoría incluyen un generador de informes. También pueden incluir un módulo gráfico que permita presentar la información con gráficos y tablas.

Hay muchos tipos distintos según cómo manejen los datos y muchos tamaños distintos de acuerdo a si operan en computadoras personales y con poca memoria o grandes sistemas que funcionan en mainframes con sistemas de almacenamiento especiales [7].

Generalmente se accede a los datos mediante lenguajes de interrogación, lenguajes de alto nivel que simplifican la tarea de construir las aplicaciones. También simplifican la interrogación y la presentación de la información. Un SGBD permite controlar el acceso a los datos, asegurar su integridad, gestionar el acceso concurrente a ellos, recuperar los datos tras un fallo del sistema y hacer copias de seguridad. Las bases de datos y los sistemas para su gestión son esenciales para cualquier área de negocio, y deben ser gestionados con esmero.

En el caso de la aplicación desarrollada en este proyecto, el sistema gestor de base de datos utilizado es Microsoft SQL Server 2014, que es un sistema de manejo de bases de datos del modelo relacional, desarrollado por la empresa Microsoft.

El lenguaje de desarrollo utilizado (por línea de comandos o mediante la interfaz gráfica de management studio) es Transact-SQL (TSQL), una implementación del estándar ANSI del lenguaje SQL, utilizado para manipular y recuperar datos (DML), crear tablas y definir relaciones entre ellas (DDL).

Metodología de desarrollo

¿Qué es Scrum?

Scrum es una metodología ágil para el desarrollo de software o la gestión de proyectos. Antes de la definición de Scrum, es imprescindible entender el concepto de ágil. El desarrollo de software ágil se define como [8]:

Generalmente, el desarrollo de software es una actividad caótica, a menudo se caracteriza por la frase “código y arreglar”. El problema del desarrollo de software reside en que el código se escribe muchas veces sin seguir un plan subyacente, y el propio diseño del sistema se improvisa a partir de muchas decisiones a corto plazo. De hecho, esto funciona bastante bien cuando el sistema es pequeño, pero conforme el sistema va creciendo, se va haciendo más difícil añadir nuevas funcionalidades. Además, también se hacen predominantes los bugs y aumenta la dificultad de arreglarlos. Para evitarlos, es necesario una larga fase de pruebas una vez que se han definido todas las funciones del sistema. Esta fase causa estragos en los cronogramas, ya que la realización de pruebas y la depuración es imposible de planificar.

El movimiento original de intentar cambiar esto, introdujo la noción de metodología. Estas metodologías imponen un proceso disciplinado sobre el desarrollo de software con el objetivo de elaborar un software más predecible y eficiente. Esto se consigue desarrollando un proceso detallado con especial énfasis en la planificación, inspirada en otras disciplinas de la ingeniería - también llamadas metodologías de la ingeniería.

Las metodologías ágiles surgieron como reacción a estas metodologías. La atracción que sufre mucha gente por estas metodologías viene provocada por la burocracia que conlleva las metodologías de la ingeniería. Estos métodos nuevos intentan ofrecer un compromiso eficiente entre procesos, proporcionando los procesos únicamente necesarios para lograr una recompensa razonable. El resultado de todo esto son los significativos cambios que sufren los métodos ágiles frente a los métodos de la ingeniería. La diferencia más determinante es que estas metodologías se basan en la poca documentación, normalmente resaltando la poca cantidad de documentos para una tarea. Se podría decir que están enfocadas al código: siguiendo una ruta que diga que la parte principal de la documentación es el código fuente.

Las metodologías ágiles son flexibles más que predictivas. Los métodos de la ingeniería tienden a intentar planificar detalladamente una gran parte del

proceso del software durante un largo periodo de tiempo, lo que funciona hasta que algo cambia. En cambio, las metodologías ágiles se acogen al cambio continuo. Intentan ser procesos que se adaptan y prosperan con el propio cambio.

Por otro lado, las metodologías ágiles están más enfocadas al ser humano que a los procesos. El objetivo de las metodologías de la ingeniería es definir un proceso que funcionará bien sin importar quién lo vaya a usar. En cambio, las metodologías ágiles reivindican que el no usar procesos supondrá el desarrollo de las habilidades del equipo, de forma que, el papel de un proceso es dar soporte al equipo desarrollador en su trabajo.

Actualmente hay muchas metodologías ágiles en uso, pero ha sido necesario el uso de esta metodología para el desarrollo del presente proyecto. Por lo tanto, es imprescindible hablar de Scrum antes de meterse en la implementación del mismo.

A continuación se presenta una breve descripción de la metodología Scrum:

La metodología Scrum para el desarrollo ágil de software representa un punto de partida de la gestión en cascada. De hecho, Scrum y otro tipo de procesos ágiles se inspiraron en sus limitaciones. La metodología Scrum enfatiza en la comunicación y colaboración, el funcionamiento del software, y la flexibilidad de la que dispone para adaptarse a las emergentes realidades de las empresas - todos los atributos de los que carecía el modelo de cascada.

¿Qué caracteriza a Scrum?

De todas las metodologías ágiles, Scrum es única porque introduce la idea del control empírico de los procesos. Esto significa que Scrum utiliza el progreso real de un proyecto para planificar y concertar los lanzamientos. En Scrum, los proyectos se dividen en ritmos de trabajo breves, conocidos como sprints. Normalmente, tienen una, dos o tres semanas de duración. Al final de cada sprint, el cliente y los miembros del equipo se reúnen para evaluar el progreso del proyecto y planear los siguientes pasos a seguir. Esto permite que la dirección del proyecto se ajuste o se reoriente una vez finalizado el trabajo, sin especulaciones ni predicciones [9].

Filosóficamente, este énfasis continuo de evaluar las tareas finalizadas es el principal causante del éxito que tiene esta metodología entre los directores y desarrolladores. Pero lo que verdaderamente permite funcionar a la metodología Scrum es un conjunto de papeles, responsabilidades, y reuniones.

Los papeles de Scrum

Scrum tiene tres papeles fundamentales: Product Owner (propietario del producto), Scrum Master (especialista en Scrum) y Team Member (miembros del equipo) [10].

- **Product Owner:** En Scrum, el Product Owner se encarga de comunicar la visión del producto al equipo de desarrollo. Él/ella también deben representar el interés del cliente por medio de los requisitos y la priorización. Como el Product Owner es el que más autoridad tiene de los tres papeles en Scrum, también es el que mayor responsabilidad recibe. En otras palabras, el Product Owner es el individuo que tiene afrontar las consecuencias cuando un proyecto va mal.

La autoridad y responsabilidad del Product Owner le hace difícil conseguir el balance correcto de implicación. Como Scrum evalúa la auto-organización entre equipos, el Product Owner debe luchar por no supervisar hasta el último detalle. Al mismo tiempo, el Product Owner tiene que estar a la disposición de las preguntas del equipo.

- **Scrum Master:** El Scrum Master actúa como enlace entre el Product Owner y el equipo. El Scrum Master no dirige al equipo. Él/ella se encarga de evitar cualquier barrera que impida al equipo lograr sus objetivos de sprint. En resumen, este papel hace que el equipo sea creativo y productivo, a la vez que permite que los logros del equipo sean visibles ante el Product Owner. El Scrum Master también aconseja al Product Owner sobre cómo maximizar el ROI (*Return Of Investment*) para el equipo.
- **Team Member:** En la metodología Scrum, el equipo es el responsable de terminar el trabajo. Idealmente, los equipos están formados por siete miembros multifuncionales, más/menos dos personas. Para proyectos de software, un equipo habitual sería una mezcla de ingenieros de software, arquitectos, programadores, analistas, testers, y diseñadores de UIs. En

cada sprint, el equipo es responsable de determinar cómo va a lograr acabar el trabajo. Esto garantiza al equipo un grado de autonomía, pero, al igual que pasa con la situación del Product Owner, esta libertad viene acompañada por la responsabilidad de cumplir los objetivos del sprint.

La siguiente imagen muestra el flujo habitual de Scrum:

1. Diagrama sencillo del proceso:

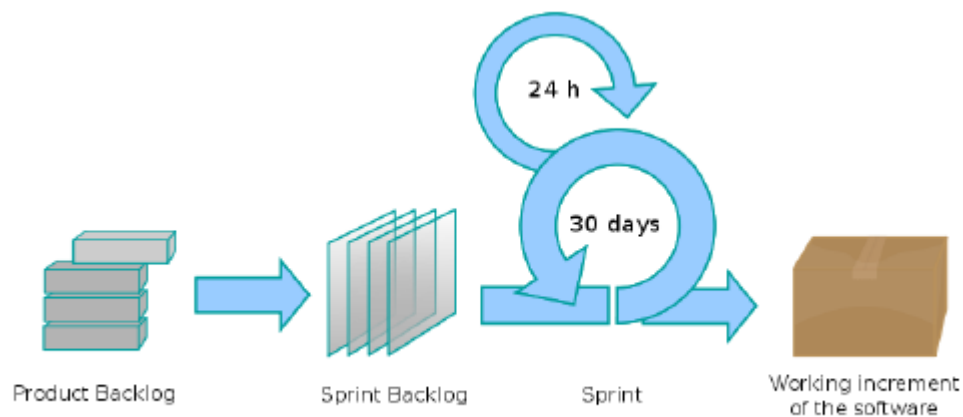


Ilustración 4 - Procesos Scrum

2. Diagrama más complejo e informativo del proceso:

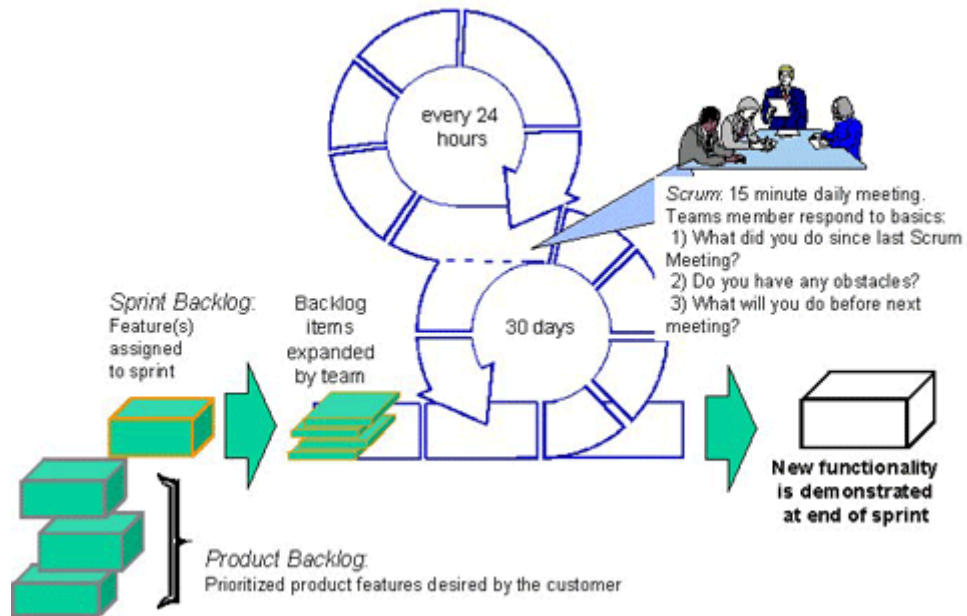


Ilustración 5 - Procesos Scrum (complejo)

Capítulo 3: Análisis del sistema

En este apartado, la meta a conseguir es identificar los objetivos que han de ser alcanzados para el desarrollo del sistema final. Para lograr esto se tomará como referencia las historias de usuario que representan las necesidades que deben cubrir las funcionalidades de la aplicación, y de este modo satisfacer las exigencias del cliente.

Para la realización del proceso de extracción de información relativa a las necesidades a cubrir se llevará a cabo entre los miembros del equipo de desarrollo y el propio cliente. Este proceso, gracias a la metodología Scrum, no se limitará a la fase inicial del proyecto, como ocurriría en un desarrollo clásico generalmente, si no que se tratará de un proceso iterativo en constante evolución para poder amoldarse a los requerimientos del cliente de la forma más eficiente.

Las historias de usuario que se extraigan de las reuniones con el cliente, describirán una funcionalidad que debe incorporar un sistema de software, y cuya implementación aporta valor al cliente.

La estructura de una historia de usuario está formada por:

- Nombre breve y descriptivo.
- Descripción de la funcionalidad en forma de diálogo o monólogo del usuario describiendo la funcionalidad que desea realizar.
- Criterio de validación y verificación que determinará para considerar terminado y aceptable por el cliente el desarrollo de la funcionalidad descrita.

Las historias de usuario se descomponen en tareas. Estas tareas son unidades de trabajo que tienen asignado un esfuerzo estimado y un estado. Por

lo que es en las tareas en los que se basa la estimación de esfuerzos general del proyecto.

Definición

Para la definición del sistema, es decir, para la extracción de la información que ayude a modelar la aplicación según las necesidades del cliente, se realiza una reunión entre todos los miembros interesados en el proyecto. En dicha reunión el cliente expone su idea del sistema, que necesidades debe cubrir y que funciones quiere que le aporte. También los miembros del equipo técnico aportan su visión para enriquecer la definición final del sistema a desarrollar. Finalmente con toda la información extraída de la reunión el Product Owner, el actor que representa la voz del cliente, escribe las historias de usuario donde se representa formalmente la definición del sistema, además se extraen las tareas que componen las historias de usuario, se les asigna un coste (un esfuerzo estimado) y las prioriza. Quedando formado el Product Backlog que será la base del proyecto a desarrollar.

Es importante aclarar que debido a la naturaleza ágil del proyecto, la definición inicial del sistema puede verse alterada en el transcurso del desarrollo del mismo. Ya que como más adelante se verá, el desarrollo se divide en diferentes etapas o Sprints, entre los cuales se repetirá la reunión entre los miembros del proyecto, pudiéndose incluir, modificar o eliminar historias de usuario y/o tareas. Por este motivo el análisis expuesto en este capítulo únicamente se trata del análisis inicial, y en próximos capítulos se describirán los análisis correspondientes a la reunión de cada Sprint.

Historias de usuario

Las historias de usuario se definirán utilizando el siguiente modelo:

Historia de Usuario	
ID	HUXX
Nombre	
Prioridad	
Riesgo	
Descripción	
Validación	

Donde cada campo tiene el siguiente significado:

- **ID:**
Se trata del identificador único asignado a este elemento del proyecto, se seguirá el formato HUXX para las historias de usuario.
- **Nombre:**
Es nombre corto utilizado para describir muy brevemente la historia de usuario.
- **Prioridad:**
Es la preferencia de cara al desarrollo de la historia de usuario respecto a las demás.
Valores: Alta, media y baja.
- **Riesgo:**
Se trata de la importancia de la historia de usuario en relación al conjunto del proyecto. Cuantificando de este modo el daño provocado en caso de fallo.
Valores: Alto, medio y bajo.

- **Descripción:**
Breve explicación de las intenciones de la historia de usuario. Debe dejar clara la idea de la propia historia.
- **Validación:**
Son las condiciones que deben cumplirse una vez la historia está completamente desarrollada para que se pueda por finalizada.

Historia de Usuario	
ID	HU01
Nombre	Introducir ventas realizadas.
Prioridad	Media
Riesgo	Bajo
Descripción	Como gestor de ventas quiero introducir las operaciones de venta de artículos realizadas en el sistema.
Validación	<ul style="list-style-type: none"> - Quiero que la operación quede registrada a nombre de un empleado. - Quiero que la operación se diferencie por centro y cliente. - Quiero que cada venta de un tipo de artículo se haga por separado. - Quiero que en la venta se almacene el precio del artículo en ese momento.

Historia de Usuario	
ID	HU02
Nombre	Introducir compras realizadas.
Prioridad	Media
Riesgo	Bajo
Descripción	Como gestor de compras quiero introducir las operaciones de compra de

	artículos realizadas en el sistema.
Validación	<ul style="list-style-type: none"> - Quiero que la compra se guarde en el sistema. - Quiero que la compra guarde una relación de proveedor, empleado que la realiza, el centro de destino y el precio pagado por la compra.

Historia de Usuario	
ID	HU03
Nombre	Registrar y administrar artículos.
Prioridad	Alta
Riesgo	Alto
Descripción	Como gestor de compras quiero registrar los artículos nuevos en el sistema.
Validación	<ul style="list-style-type: none"> - Quiero poner introducir nuevos artículos en el sistema. - Los artículos deben agruparse por su tipología. - Cada artículo tiene asociado un precio de venta. - Los datos almacenados se pueden modificar posteriormente.

Historia de Usuario	
ID	HU04
Nombre	Registrar y administrar proveedores.
Prioridad	Alta
Riesgo	Alto
Descripción	Como gestor de compras quiero registrar nuevos proveedores.
Validación	<ul style="list-style-type: none"> - Quiero poder introducir nuevos proveedores en el sistema. - Cada proveedor debe tener relacionado un nombre, un NIF, un email y una cuenta bancaria. - Puede haber almacenados dos proveedores con el mismo NIF. - Los datos almacenados se pueden modificar posteriormente.

--	--

Historia de Usuario	
ID	HU05
Nombre	Registrar y administrar centros.
Prioridad	Alta
Riesgo	Alto
Descripción	Como gestor de almacenes quiero disponer de diferentes centros como destino / origen de las mercancías.
Validación	<ul style="list-style-type: none"> - Quiero poder introducir nuevos centros en el sistema. - Cada centro debe tener relacionado un nombre, un teléfono, un email y una dirección física. - Los datos almacenados se pueden modificar posteriormente.

Historia de Usuario	
ID	HU06
Nombre	Control de stocks.
Prioridad	Alta
Riesgo	Medio
Descripción	Como gestor de almacenes quiero tener control de los stocks de los artículos que hay en los centros.
Validación	<ul style="list-style-type: none"> - Quiero incluir nuevos artículos a un centro. - Quiero aumentar o reducir la cantidad almacenada. - Quiero poder tener el mismo artículo en varios centros.

Historia de

Usuario	
ID	HU07
Nombre	Apariencia de la aplicación
Prioridad	Baja
Riesgo	Bajo
Descripción	Como usuario quiero que la aplicación tenga un aspecto simple y sencillo de manejar.
Validación	<ul style="list-style-type: none"> - Quiero que las ventanas sean reducidas y contengan solo la información imprescindible. - Quiero que para entrar en algún modo se abra una ventana específica. - Quiero unas funciones básicas (comprar y vender) de fácil acceso.

Historia de Usuario	
ID	HU08
Nombre	Funcionamiento en Windows
Prioridad	Alta
Riesgo	Alto
Descripción	Como usuario quiero que el sistema funcione en el sistema operativo Windows 8.1
Validación	<ul style="list-style-type: none"> - Quiero que la aplicación funcione en Windows 8.1 - Diseño de la arquitectura de la aplicación.

Historia de Usuario	
ID	HU09
Nombre	Importación / Exportación desde Excel
Prioridad	Baja
Riesgo	Bajo
Descripción	Como usuario quiero introducir y extraer datos en formato Excel.

Validación	<ul style="list-style-type: none"> - Quiero importar datos desde un fichero .xls. - Quiero exportar datos a un fichero .xls.

Historia de Usuario	
ID	HU10
Nombre	Base de datos
Prioridad	Alta
Riesgo	Alto
Descripción	Como desarrollador quiero que los datos introducidos sean persistentes.
Validación	<ul style="list-style-type: none"> - Quiero el uso de una base de datos. - Quiero que a la base de datos se le puedan introducir nuevos datos. - Quiero que los datos de la base de datos se puedan modificar. - Quiero que los datos de la base de datos se puedan eliminar.

Tareas

Las tareas que componen las historias de usuario se detallan a continuación siguiendo el siguiente modelo:

Tarea	Txx
Historia de Usuario	HUxx

Estado	
Descripción	

Donde cada campo tiene el siguiente significado:

- **Tarea:**
Es el identificador único para este elemento del proyecto. El formato que sigue es Txx, donde las “xx” serán un número entero del 01 al 99.
- **Historia de Usuario:**
Toda tarea forma parte de una historia de usuario que se indica en este campo.
- **Estado:**
Se trata de la fase en la que se encuentra el desarrollo de la tarea.
Valores: No iniciada, En progreso y Completada.
- **Descripción:**
Una breve explicación de la finalidad de la tarea.

Tarea	T01
Historia de Usuario	HU03
Estado	
Descripción	Crear interfaz con la BBDD para la tabla artículos en la capa de datos. DArticulos.cs

Tarea	T02
Historia de Usuario	HU03
Estado	
Descripción	Crear NArticulos.cs en la capa de negocio.

Tarea	T03
Historia de Usuario	HU03
Estado	
Descripción	Crear tabla artículos en la base de datos.

Tarea	T04
Historia de Usuario	HU03
Estado	
Descripción	Crear procedimientos para consulta e inserción en la tabla artículos.

Tarea	T05
Historia de Usuario	HU03
Estado	
Descripción	Crear tabla grupo_articulos en la base de datos.

Tarea	T06
Historia de Usuario	HU03
Estado	
Descripción	Crear procedimientos para consulta e inserción en la tabla grupo_artículos.

Tarea	T07
Historia de Usuario	HU03
Estado	
Descripción	Crear interfaz con la BBDD para la tabla artículos en la capa de datos. DGrupo_Articulos.cs

Tarea	T08
Historia de Usuario	HU03
Estado	
Descripción	Crear NArticulos.cs en la capa de negocio.

Tarea	T09
Historia de Usuario	HU03
Estado	
Descripción	Crear la interfaz de usuario para la gestión de la tabla artículos.

Tarea	T10
Historia de Usuario	HU03
Estado	
Descripción	Crear la interfaz de usuario para la gestión de la tabla grupo_artículos.

Tarea	T11
Historia de Usuario	HU05
Estado	
Descripción	Crear la tabla centros en la base de datos.

Tarea	T12
Historia de Usuario	HU05
Estado	
Descripción	Crear los procedimientos en la base de datos para las consultas, inserciones y modificaciones en la tabla centros.

Tarea	T13
Historia de Usuario	HU05
Estado	
Descripción	Crear DCentros.cs en la capa de datos.

Tarea	T14
Historia de Usuario	HU05
Estado	
Descripción	Crear NCentros.cs en la capa de negocio.

Tarea	T15
Historia de Usuario	HU05
Estado	
Descripción	Crear la interfaz de usuario para la gestión de la tabla centros.

Tarea	T16
Historia de Usuario	HU06
Estado	
Descripción	Crear la tabla stocks en la base de datos.

Tarea	T17
Historia de Usuario	HU06
Estado	
Descripción	Crear los procedimientos en la base de datos para las consultas, inserciones y modificaciones en la tabla stocks.

Tarea	T18
Historia de Usuario	HU06
Estado	
Descripción	Crear DStocks.cs en la capa de datos.

Tarea	T19
Historia de Usuario	HU06
Estado	
Descripción	Crear NStocks.cs en la capa de negocio.

Tarea	T20
Historia de Usuario	HU06
Estado	
Descripción	Crear la interfaz de usuario para la gestión de la tabla stocks.

Tarea	T21
Historia de Usuario	HU06
Estado	
Descripción	Controlar que no hay stocks negativos.

Tarea	T22
Historia de Usuario	HU07
Estado	
Descripción	Diseño de la interfaz gráfica de la aplicación.

Tarea	T23
Historia de Usuario	HU10
Estado	0.5
Descripción	Determinar tablas de la base de datos.

Tarea	T24
Historia de Usuario	HU10
Estado	0.5
Descripción	Diseño de relaciones entre las tablas de la base de datos.

Tarea	T25
Historia de Usuario	HU04
Estado	
Descripción	Crear la tabla proveedores en la base de datos.

Tarea	T26
Historia de Usuario	HU04
Estado	
Descripción	Crear los procedimientos en la base de datos para las consultas, inserciones y modificaciones en la tabla proveedores.

Tarea	T27
Historia de Usuario	HU04
Estado	
Descripción	Crear DProveedores.cs en la capa de datos.

Tarea	T28
Historia de Usuario	HU06
Estado	
Descripción	Crear NProveedores.cs en la capa de negocio.

Tarea	T29
Historia de Usuario	HU06
Estado	
Descripción	Crear la interfaz de usuario para la gestión de la tabla proveedores.

Tarea	T30
Historia de Usuario	HU01
Estado	
Descripción	Crear NVentas.cs en la capa de negocio.

Tarea	T31
Historia de Usuario	HU01
Estado	
Descripción	Crear la interfaz de usuario para la gestión de las ventas.

Tarea	T32
Historia de Usuario	HU01
Estado	
Descripción	Crear DVentas.cs en la capa de datos.

Tarea	T33
Historia de Usuario	HU02
Estado	
Descripción	Crear NCompras.cs en la capa de negocio.

Tarea	T34
Historia de Usuario	HU02
Estado	
Descripción	Crear la interfaz de usuario para la gestión de las compras.

Tarea	T35
Historia de Usuario	HU02
Estado	
Descripción	Crear DCompras.cs en la capa de datos.

Tarea	T36
Historia de Usuario	HU02
Estado	
Descripción	Crear la interfaz de usuario para la gestión de las compras.

Tarea	T37
Historia de Usuario	HU09
Estado	
Descripción	Crear plantilla de Excel para cada tabla.

Tarea	T38
Historia de Usuario	HU09
Estado	

Descripción	Crear DExcel.cs en la capa de datos.
--------------------	--------------------------------------

Tarea	T39
Historia de Usuario	HU09
Estado	
Descripción	Crear la interfaz de usuario para la importación de datos desde Excel.

Tarea	T40
Historia de Usuario	HU08
Estado	
Descripción	Diseño de la arquitectura del sistema

Tarea	T41
Historia de Usuario	HU01
Estado	
Descripción	Crear NCliente.cs en la capa de negocio.

Tarea	T42
Historia de Usuario	HU01
Estado	
Descripción	Crear la interfaz de usuario para la gestión de los clientes.

Tarea	T43
Historia de Usuario	HU01
Estado	
Descripción	Crear DClientes.cs en la capa de datos.

Tarea	T44
Historia de Usuario	HU01

Estado	
Descripción	Crear la tabla clientes en la base de datos.

Tarea	T45
Historia de Usuario	HU01
Estado	
Descripción	Crear los procedimientos en la base de datos para las consultas, inserciones y modificaciones en la tabla clientes.

Tarea	T46
Historia de Usuario	HU01
Estado	
Descripción	Crear NEmpleados.cs en la capa de negocio.

Tarea	T47
Historia de Usuario	HU01
Estado	
Descripción	Crear la interfaz de usuario para la gestión de los empleados.

Tarea	T48
Historia de Usuario	HU01
Estado	
Descripción	Crear DEmpleados.cs en la capa de datos.

Tarea	T49
Historia de Usuario	HU01
Estado	
Descripción	Crear la tabla empleados en la base de datos.

Tarea	T50
--------------	------------

Historia de Usuario	HU01
Estado	
Descripción	Crear los procedimientos en la base de datos para las consultas, inserciones y modificaciones en la tabla empleados.

Estimación

La estimación es el proceso mediante el cual se obtiene una previsión del esfuerzo necesario para desarrollar las diferentes partes del proyecto. Esta información será de gran utilidad a la hora de organizar el desarrollo del producto.

La estimación de proyectos software es una tarea muy compleja, pero de vital importancia en toda la etapa de desarrollo del software. Se ha de tener en cuenta que las estimaciones realizadas nunca serán del todo precisas por la gran cantidad de variables que afectan en su cálculo. Pero, obviamente, cuanto mejor sea la estimación que se haga, mejor se podrá maniobrar para lograr cumplir los objetivos y mejorar la rentabilidad del proyecto.

En el caso de este proyecto concreto, se está tomando como referencia la metodología ágil Scrum. Para las metodologías ágiles existe una técnica bastante extendida para la estimación. Se trata de la técnica de Planning Poker. En el siguiente apartado se describe el funcionamiento de este método para estimar.

Además esta técnica se puede utilizar conjuntamente a otros métodos como el juicio de un experto y las estadísticas de rendimiento del equipo en proyectos similares. En caso de utilizar la combinación de varias técnicas habría que asignar un peso a cada una de ellas y utilizar una fórmula de este estilo, ejemplo:

Juicio experto: 20%

Planning Poker: 40%

Estadísticas de Velocidad del equipo: 40%

El resultado final para esta historia sería: $20\%(20) + 40\%(15) + 40\%(11)$; es decir 14

En el caso concreto de este proyecto se utilizará únicamente Planning Poker debido a la falta de un experto que pueda aportar su opinión y también a la falta de experiencia del equipo en proyectos similares, por lo que no habría nada en lo que basarse.

Planning Poker

Una de las técnicas más extendidas para la ayuda de la estimación de las historias de usuario en Scrum es la conocida como Planning Poker. Este proceso tiene como objetivo el poder realizar un cálculo del esfuerzo necesario para llevar a cabo las distintas historias de usuario mediante un consenso entre los miembros encargados de realizar las distintas tareas que componen cada fase.

La técnica consiste en que cada miembro del equipo elige una carta de la baraja que representa el valor estimado del esfuerzo que él considera para la tarea sin mostrársela al resto de los miembros del equipo para no influenciar su decisión. Cuando todos los miembros del equipo han elegido su carta, la muestran al resto para así elegir la estimación final según la mayoría. En caso de discrepancias muy altas se puede realizar un pequeño debate para ver los puntos de discordia y llegar a un acuerdo. Los valores de la baraja suelen ser la secuencia de Fibonacci, con alguna pequeña variación, por lo que no es una sucesión lineal, de esta forma se refleja la incertidumbre a la hora de realizar las estimaciones, a mayor grado de complejidad, mayor es esta incertidumbre.

Esta técnica sirve como se ha mencionado anteriormente para la estimación de historias de usuario, es decir el conjunto de todas las tareas de que la componen, en lugar de estimar cada tarea de forma individualizada. Ambas técnicas son aceptables y cuentan con numerosos seguidores, pero en el caso de este proyecto se ha optado por la estimación de historias de usuario completas frente a la estimación de tareas, puesto que se ha considerado esta como la forma más óptima de realizar el proceso de estimación. Un estudio realizado [11] encontró que las estimaciones obtenidas mediante el proceso de Planificación de póker son menos optimistas y más precisa que las estimaciones obtenidas a través de la combinación mecánica de las estimaciones individuales en las mismas tareas.

Estimación en el proyecto

En este apartado se explicará la forma que se estimará este proyecto después del análisis para la extracción de las historias de usuario y sus tareas.

El método que se va a utilizar estará basado en el Planning Poker anteriormente explicado, pero adaptado a las condiciones especiales de este proyecto. Esta adaptación será lo que se detalle en este capítulo.

Para realizar la estimación de las historias de usuario se utilizará la técnica de Planning Poker al inicio de cada iteración (sprint), en donde se aplicará a todas las historias de usuario restantes. De este modo se ordenarán por prioridad y se obtendrá como resultado la cantidad de trabajo que se llevará a cabo durante ese sprint.

Hay que tener en cuenta que el prototipo resultante de una iteración debe ser funcional en medida de sus posibilidades, por lo que a la hora de asignar las historias de usuario a realizar en el sprint siempre debe ser inferior la suma de tiempos estimados que el tiempo total de la iteración. Así, en un principio se evitaría que quedase una historia a medio hacer afectando al funcionamiento del prototipo.

Para priorizar las historias de usuario, al principio del proyecto, y a intervalos regulares durante el proyecto, se va estimando el esfuerzo de cada una en puntos de usuario. Normalmente se utilizan los números **0, 1, 3, 5, 8, 13, 20, 40 y 100** como puntos de usuario. 0 indica una historia con esfuerzo prácticamente nulo, mientras que una con 100, indica que el esfuerzo es extremo y que incluso no es asumible ahora mismo. El Product Owner expone cada una de las historias y el equipo intenta llegar a un consenso con el esfuerzo asignado. Se suelen utilizar otras historias para comparar. Estos valores serán los utilizados en este proyecto.

Un ejemplo de funcionamiento y de formato de las tablas de estimación utilizando Planning Poker es la siguiente, que además será el utilizado durante el desarrollo del presente proyecto.

Historia de Usuario	Miembro			Estimación media	Prioridad
	A	B	C		
HU01	13	20	13	15,33333333	Baja
HU02	13	13	20	15,33333333	Alta
HU03	5	8	8	7	Media

En la tabla del Planning Poker podemos observar las valoraciones de cada miembro del equipo que se ha supuesto para el desarrollo del proyecto. Se observa como han utilizado valores para las horas de la serie **0, 1, 3, 5, 8, 13, 20, 40 y 100**. De este modo lo primero sería establecer la capacidad de trabajo del

equipo durante una iteración. Para este ejemplo se va a suponer una capacidad de trabajo de 20 horas.

La solución sería “rellenar” ese espacio disponible de 20 horas con las historias de usuario por completar, teniendo en cuenta lo prioritario de cada una de estas, por lo que se deben coger primero las de prioridad alta, después prioridad media y por último prioridad baja.

Según el ejemplo anterior en este primer sprint se llevaría a cabo la historia de usuario 2, que tiene una estimación de 15,3 horas, puesto que la siguiente historia de usuario no tendría cabida en este sprint.

El tiempo que existe de diferencia entre el estimado y el total de la iteración puede utilizarse para solventar los normales errores de estimación, permitiendo recuperar posibles retrasos en el desarrollo debidos a cualquier causa. Lo importante es que al finalizar el sprint el resultado sea un prototipo funcional que pueda ser presentado al cliente.

Capítulo 4: Desarrollo del sistema

En este capítulo se realizará el seguimiento del desarrollo del proyecto, desde el fin del análisis hasta completar todas las historias de usuario detalladas en este análisis del sistema.

Se va a distribuir en varias partes. Primeramente una visión global del desarrollo, donde se explicará las tecnologías utilizadas y el motivo de su elección. Luego se detallará cada fase del desarrollo, llamada Sprint. En total se contará con seis Sprint de dos semanas cada uno. Más adelante en el apartado de los Sprint se describirá y explicará con más detalle el funcionamiento y composición de éstos.

Visión general

Para el desarrollo del sistema se ha seleccionado una combinación de tecnologías y metodologías que incluyen las explicadas en el capítulo de “Estado del arte”. En ese capítulo se exponen diferentes posibilidades en algunos casos como el lenguaje o las herramientas de desarrollo del proyecto. La metodología a utilizar está ya predeterminada por la naturaleza del proyecto, siendo imprescindible el empleo de Scrum por este motivo.

La combinación elegida para el desarrollo es:

- Plataforma .NET
 - Microsoft Visual Studio 2013.
 - Lenguaje C#.
 - Microsoft SQL Server 2014.

En la historia de usuario HU08 se pide que la aplicación debe funcionar en Windows, por lo que de entre las opciones disponibles, se ha considerado que esta combinación de productos Microsoft es la más óptima.

Las ventajas de esta configuración son la fiabilidad de una aplicación desarrollada en esta plataforma para Windows, las facilidades que proporciona

Visual Studio para el desarrollo de una interfaz sencilla y a la vez bien integrada en el sistema operativo Windows.

Otro beneficio de dicha combinación es la sencilla conexión con la base de datos.

Arquitectura

El modelo de arquitectura que se ha seleccionado para el desarrollo de la aplicación del proyecto está basado en un modelo Cliente – Servidor.

Cliente - Servidor

Es un modelo para construir sistemas de información, que se sustenta en la idea de repartir el tratamiento de la información y los datos por todo el sistema informático, permitiendo mejorar el rendimiento del sistema global de información. A continuación se muestra una ilustración de este sistema:

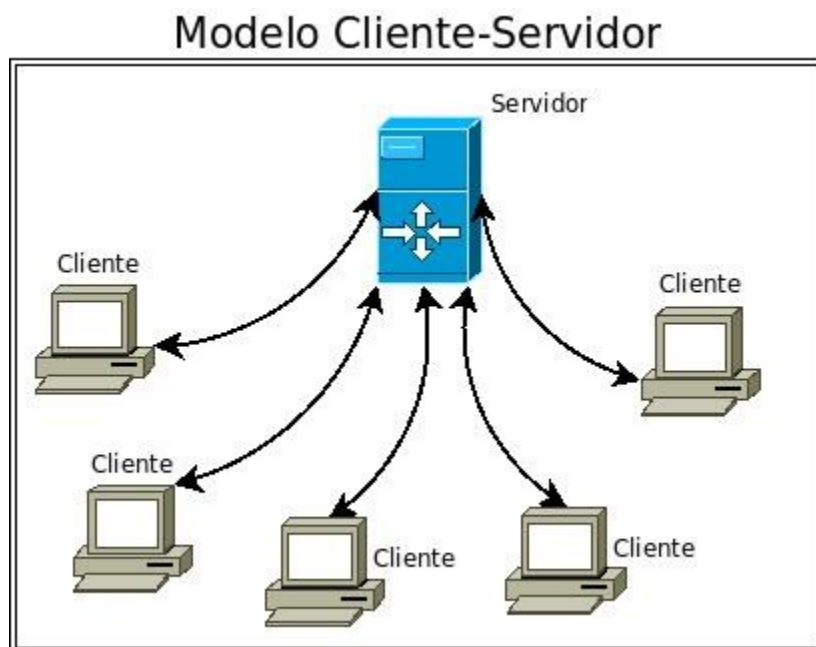


Ilustración 6 - Cliente-Servidor

En el caso de este proyecto, en el lado del servidor reside toda la información de la base de datos, además de que es donde corre el sistema gestor de la base de datos (SGDB), en este caso Microsoft SQL Server 2014.

Por otra parte se encuentra el lado del cliente, el cual sería la aplicación que utiliza el usuario para introducir, modificar o visualizar los datos almacenados en el sistema.

MVC

La aplicación del lado del cliente, se ha estructurado siguiendo el patrón Modelo-Vista-Controlador (MVC).

El modelo–vista–controlador (MVC) es un patrón de arquitectura de software que separa los datos y la lógica de negocio de una aplicación de la interfaz de usuario y el módulo encargado de gestionar los eventos y las comunicaciones. Para ello MVC propone la construcción de tres componentes distintos que son el modelo, la vista y el controlador, es decir, por un lado define componentes para la representación de la información, y por otro lado para la interacción del usuario. Este patrón de arquitectura de software se basa en las ideas de reutilización de código y la separación de conceptos, características que buscan facilitar la tarea de desarrollo de aplicaciones y su posterior mantenimiento [12].

De manera genérica, los componentes de MVC se podrían definir como sigue:

- **Modelo:** Es la representación de la información con la cual el sistema opera, por lo tanto gestiona todos los accesos a dicha información, tanto consultas como actualizaciones, implementando también los privilegios de acceso que se hayan descrito en las especificaciones de la aplicación (lógica de negocio). Envía a la 'vista' aquella parte de la información que en cada momento se le solicita para que sea mostrada (típicamente a un usuario). Las peticiones de acceso o manipulación de información llegan al 'modelo' a través del 'controlador'.
- **Controlador:** Responde a eventos (usualmente acciones del usuario) e invoca peticiones al 'modelo' cuando se hace alguna solicitud sobre la

información (por ejemplo, editar un documento o un registro en una base de datos). También puede enviar comandos a su 'vista' asociada si se solicita un cambio en la forma en que se presenta el 'modelo' (por ejemplo, desplazamiento o scroll por un documento o por los diferentes registros de una base de datos), por tanto se podría decir que el 'controlador' hace de intermediario entre la 'vista' y el 'modelo'.

- **Vista:** Presenta el 'modelo' (información y lógica de negocio) en un formato adecuado para interactuar (usualmente la interfaz de usuario) por tanto requiere de dicho 'modelo' la información que debe representar como salida.

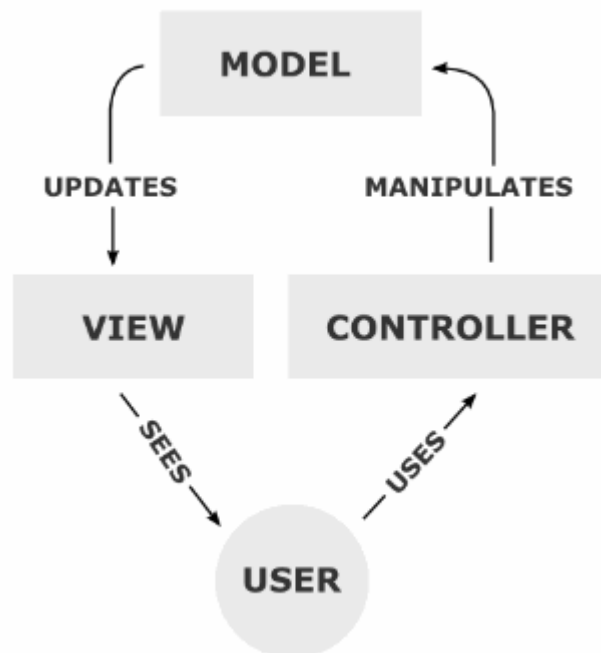


Ilustración 7 - MVC

Aunque se pueden encontrar diferentes implementaciones de **MVC**, el flujo de control que se sigue generalmente es el siguiente:

1. El usuario interactúa con la interfaz de usuario de alguna forma (por ejemplo, el usuario pulsa un botón, enlace, etc.)
2. El controlador recibe (por parte de los objetos de la interfaz-vista) la notificación de la acción solicitada por el usuario. El controlador gestiona el evento que llega, frecuentemente a través de un gestor de eventos (handler) o callback.

3. El controlador accede al modelo, actualizándolo, posiblemente modificándolo de forma adecuada a la acción solicitada por el usuario (por ejemplo, el controlador actualiza el carro de la compra del usuario). Los controladores complejos están a menudo estructurados usando un patrón de comando que encapsula las acciones y simplifica su extensión.
4. El controlador delega a los objetos de la vista la tarea de desplegar la interfaz de usuario. La vista obtiene sus datos del modelo para generar la interfaz apropiada para el usuario donde se reflejan los cambios en el modelo (por ejemplo, produce un listado del contenido del inventario). El modelo no debe tener conocimiento directo sobre la vista. Sin embargo, se podría utilizar el patrón Observador para proveer cierta indirección entre el modelo y la vista, permitiendo al modelo notificar a los interesados de cualquier cambio. Un objeto vista puede registrarse con el modelo y esperar a los cambios, pero aun así el modelo en sí mismo sigue sin saber nada de la vista. Este uso del patrón Observador no es posible en las aplicaciones Web puesto que las clases de la vista están desconectadas del modelo y del controlador. En general el controlador no pasa objetos de dominio (el modelo) a la vista aunque puede dar la orden a la vista para que se actualice. *Nota: En algunas implementaciones la vista no tiene acceso directo al modelo, dejando que el controlador envíe los datos del modelo a la vista, por ejemplo en el proyecto que se desarrolla.*
5. La interfaz de usuario espera nuevas interacciones del usuario, comenzando el ciclo nuevamente.

En la aplicación que se desarrolla en este proyecto el patrón modelo – vista – controlador, junto a la base de datos en el lado del servidor, quedaría estructurado de la siguiente forma que se muestra en el diagrama:

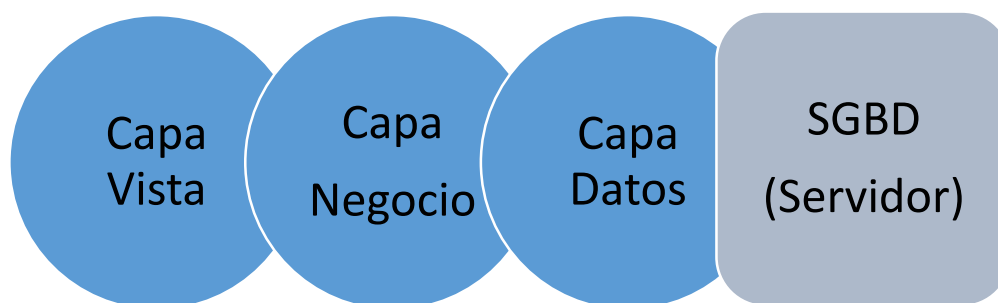


Ilustración 8 - Capas

Empezando desde el nivel superior, tenemos la vista, que se corresponde con la capa con su mismo nombre, Capa Vista. En esta capa se implementará todo lo relacionado con la interfaz de usuario, siendo esta la única con la que el usuario podrá interactuar directamente. Por otra parte la capa Vista interactuará con la Capa Negocio para obtener e insertar datos en el sistema.

Después se dispone de la Capa Negocio, que es la correspondiente al controlador del patrón MVC. En la capa Negocio se gestionará el funcionamiento del sistema. Esta capa conecta por un lado con la capa Vista y por el otro con la capa de Datos.

La última capa correspondiente al patrón MVC es la capa Datos, que representa al Modelo. Esta capa se encarga de recibir la información desde la base de datos y transformarla en formato útil para manejar desde la aplicación, y del mismo modo actúa en el sentido contrario, ajustar los datos proporcionados por la capa Negocio para ser insertados en la base de datos.

Por último se dispone del lado servidor, en el cual se sitúa el sistema gestor de base de datos, Microsoft SQL Server 2014 en este caso. En el capítulo propio se explica que es un SGBD y cuál es su función.

Como dato importante cabe resaltar que las consultas a la base de datos se realizarán mediante procedimientos. En el siguiente apartado se detallará este punto.

Capa Datos

Para la conexión de la capa de datos entre el cliente y el servidor se utilizarán procedimientos almacenados en la base de datos para la realización de consultas en esta.

Un procedimiento almacenado [13] de SQL Server es un grupo de una o varias instrucciones Transact-SQL o una referencia a un método de Common Runtime Language (CLR) de Microsoft .NET Framework. Los procedimientos se asemejan a las construcciones de otros lenguajes de programación, porque pueden:

- Aceptar parámetros de entrada y devolver varios valores en forma de parámetros de salida al programa que realiza la llamada.

- Contener instrucciones de programación que realicen operaciones en la base de datos. Entre otras, pueden contener llamadas a otros procedimientos.
- Devolver un valor de estado a un programa que realiza una llamada para indicar si la operación se ha realizado correctamente o se han producido errores, y el motivo de estos.

Ventajas de usar procedimientos almacenados

En la siguiente lista se describen algunas de las ventajas que brinda el uso de procedimientos.

- Tráfico de red reducido entre el cliente y el servidor.

Los comandos de un procedimiento se ejecutan en un único lote de código. Esto puede reducir significativamente el tráfico de red entre el servidor y el cliente porque únicamente se envía a través de la red la llamada que va a ejecutar el procedimiento. Sin la encapsulación de código que proporciona un procedimiento, cada una de las líneas de código tendría que enviarse a través de la red.

- Mayor seguridad:
 - Varios usuarios y programas cliente pueden realizar operaciones en los objetos de base de datos subyacentes a través de un procedimiento, aunque los usuarios y los programas no tengan permisos directos sobre esos objetos subyacentes. El procedimiento controla qué procesos y actividades se llevan a cabo y protege los objetos de base de datos subyacentes. Esto elimina la necesidad de conceder permisos en cada nivel de objetos y simplifica los niveles de seguridad.
 - La cláusula EXECUTE AS puede especificarse en la instrucción CREATE PROCEDURE para habilitar la suplantación de otro usuario o para permitir que los usuarios o las aplicaciones puedan realizar ciertas actividades en la base de datos sin necesidad de contar con permisos directos sobre los objetos y comandos subyacentes. Por ejemplo, algunas acciones como TRUNCATE TABLE no tienen

permisos que se puedan conceder. Para poder ejecutar TRUNCATE TABLE, el usuario debe tener permisos ALTER en la tabla especificada. Puede que la concesión de permisos ALTER a un usuario en una tabla no sea lo ideal, pues en realidad el usuario tendrá permisos muy superiores a la posibilidad de truncar una tabla. Si se incorpora la instrucción TRUNCATE TABLE en un módulo y se especifica la ejecución del módulo como un usuario con permisos para modificar la tabla, se pueden ampliar los permisos para truncar la tabla al usuario al que se concedan permisos EXECUTE para el módulo.

- Al llamar a un procedimiento a través de la red, solo está visible la llamada que va a ejecutar el procedimiento. Por lo tanto, los usuarios malintencionados no pueden ver los nombres de los objetos de base de datos y tabla, incrustados en sus propias instrucciones Transact-SQL, ni buscar datos críticos.
- El uso de parámetros de procedimientos ayuda a protegerse contra ataques por inyección de código SQL. Dado que la entrada de parámetros se trata como un valor literal y no como código ejecutable, resulta más difícil para un atacante insertar un comando en la instrucción Transact-SQL del procedimiento y comprometer la seguridad.

- Reutilización del código

El código de cualquier operación de base de datos redundante resulta un candidato perfecto para la encapsulación de procedimientos. De este modo, se elimina la necesidad de escribir de nuevo el mismo código, se reducen las inconsistencias de código y se permite que cualquier usuario o aplicación que cuente con los permisos necesarios pueda acceder al código y ejecutarlo.

- Mantenimiento más sencillo

Cuando las aplicaciones cliente llaman a procedimientos y mantienen las operaciones de base de datos en la capa de datos, solo deben actualizarse los cambios de los procesos en la base de datos subyacente. El nivel de aplicación permanece independiente y no tiene que tener conocimiento sobre los cambios realizados en los diseños, las relaciones o los procesos de la base de datos.

- Rendimiento mejorado

De forma predeterminada, un procedimiento se compila la primera vez que se ejecuta y crea un plan de ejecución que vuelve a usarse en posteriores ejecuciones. Como el procesador de consultas no tiene que crear un nuevo plan, normalmente necesita menos tiempo para procesar el procedimiento.

Si ha habido cambios importantes en las tablas o datos a los que se hace referencia en el procedimiento, el plan precompilado podría hacer que el procedimiento se ejecutara con mayor lentitud. En este caso, volver a crear el procedimiento y forzar un nuevo plan de ejecución puede mejorar el rendimiento.

Tipos de procedimientos almacenados

- Definidos por el usuario

Un procedimiento definido por el usuario se puede crear en una base de datos definida por el usuario o en todas las bases de datos del sistema excepto en la base de datos Resource. El procedimiento se puede desarrollar en Transact-SQL o como una referencia a un método de Common Runtime Language (CLR) de Microsoft .NET Framework.

- Temporales

Los procedimientos temporales son una forma de procedimientos definidos por el usuario. Los procedimientos temporales son iguales que los procedimientos permanentes salvo porque se almacenan en tempdb. Hay dos tipos de procedimientos temporales: locales y globales. Se diferencian entre sí por los nombres, la visibilidad y la disponibilidad. Los procedimientos temporales locales tienen como primer carácter de sus nombres un solo signo de número (#); solo son visibles en la conexión actual del usuario y se eliminan cuando se cierra la conexión. Los procedimientos temporales globales presentan dos signos de número (##) antes del nombre; son visibles para cualquier usuario después de su creación y se eliminan al final de la última sesión en la que se usa el procedimiento.

- Sistema

Los procedimientos del sistema se incluyen con SQL Server. Están almacenados físicamente en la base de datos interna y oculta Resource y se muestran de forma lógica en el esquema sys de cada base de datos definida por el sistema y por el usuario. Además, la base de datos msdb también contiene procedimientos almacenados del sistema en el esquema dbo que se usan para programar alertas y trabajos. Dado que los procedimientos del sistema empiezan con el prefijo sp_, le recomendamos que no use este prefijo cuando asigne un nombre a los procedimientos definidos por el usuario. Para obtener una lista completa de los procedimientos del sistema, vea Procedimientos almacenados del sistema (Transact-SQL).

SQL Server admite los procedimientos del sistema que proporcionan una interfaz de SQL Server a los programas externos para varias actividades de mantenimiento. Estos procedimientos extendidos usan el prefijo xp_. Para obtener una lista completa de los procedimientos extendidos, vea Procedimientos almacenados extendidos generales (Transact-SQL).

- Extendidos definidos por el usuario

Los procedimientos extendidos permiten crear rutinas externas en un lenguaje de programación como C. Estos procedimientos son archivos DLL que una instancia de SQL Server puede cargar y ejecutar dinámicamente.

Sprints

En Scrum un proyecto se ejecuta en bloques temporales cortos y fijos (iteraciones de un mes natural y hasta de dos semanas). Cada iteración tiene que proporcionar un resultado completo, un incremento de producto que sea potencialmente entregable, de manera que cuando el cliente (Product Owner) lo solicite sólo sea necesario un esfuerzo mínimo para que el producto esté disponible para ser utilizado. Para ello, durante la iteración el equipo colabora estrechamente y se llevan a cabo las siguientes dinámicas:

- Cada día el equipo realiza una reunión de sincronización, donde cada miembro inspecciona el trabajo de los otros para poder hacer las adaptaciones necesarias, comunica cuales son los impedimentos con que se encuentra, actualiza el estado de la lista de tareas de la iteración (Sprint Backlog) y los gráficos de trabajo pendiente (Burndown charts).
- El Facilitador (Scrum Master) se encarga de que el equipo pueda cumplir con su compromiso y de que no se merme su productividad.
 - Elimina los obstáculos que el equipo no puede resolver por sí mismo.
 - Protege al equipo de interrupciones externas que puedan afectar su compromiso o su productividad.

También existen una serie de restricciones que deben ser tenidas en consideración:

- No se puede cambiar los objetivos/requisitos de la iteración en curso:
 - En la reunión de planificación de la iteración el equipo adquirió el compromiso de completar en la iteración unos requisitos concretos, basó su plan y organización en ellos. Cambiar los objetivos/requisitos de la iteración dificulta la concentración del equipo, baja su moral y su compromiso.
 - El hecho de no poder cambiar los objetivos/requisitos de la iteración una vez iniciada facilita que el cliente cumpla con su responsabilidad de conocer qué es lo más prioritario a desarrollar, antes de iniciar la iteración.
 - Notar que Scrum minimiza esta necesidad ya que, por un lado, los objetivos/requisitos que se están desarrollando eran los más

prioritarios justo antes de iniciar la iteración y, por otro lado, las iteraciones en Scrum son suficientemente cortas (2 o 4 semanas) como para que la probabilidad de cambios una vez iniciada la iteración sea mínima.

- Terminación anormal de la iteración:
 - Sólo en situaciones muy excepcionales el cliente o el equipo pueden solicitar una terminación anormal de la iteración. Esto puede suceder si, por ejemplo, el contexto del proyecto ha cambiado enormemente y no es posible esperar al final de la iteración para aplicar cambios, o si el equipo encuentra que es imposible cumplir con el compromiso adquirido. En ese caso, se dará por finalizada la iteración y se dará inicio a otra mediante una reunión de planificación de la iteración.

Sprint Planning

La planificación de las tareas a realizar en la iteración se divide en dos partes:

- Primera parte de la reunión. Se realiza en un timebox de cómo máximo 4 horas *(aproximadamente):
 - El cliente presenta al equipo la lista de requisitos priorizada del producto o proyecto, pone nombre a la meta de la iteración (de manera que ayude a tomar decisiones durante su ejecución) y propone los requisitos más prioritarios a desarrollar en ella.
 - El equipo examina la lista, pregunta al cliente las dudas que le surgen, añade más condiciones de satisfacción y selecciona los objetivos/requisitos más prioritarios que se compromete a completar en la iteración, de manera que puedan ser entregados si el cliente lo solicita.

- Segunda parte de la reunión. Se realiza en un timebox de cómo máximo 4 horas*. El equipo planifica la iteración, elabora la táctica que le permitirá conseguir el mejor resultado posible con el mínimo esfuerzo. Esta actividad la realiza el equipo dado que ha adquirido un compromiso, es el responsable de organizar su trabajo y es quien mejor conoce cómo realizarlo.
 - Define las tareas necesarias para poder completar cada objetivo/requisito, creando la lista de tareas de la iteración (Sprint Backlog) basándose en la definición de completado.
 - Realiza una estimación conjunta del esfuerzo necesario para realizar cada tarea.
 - Cada miembro del equipo se autoasigna a las tareas que puede realizar.

Sprint 0

Se conoce como Sprint 0 a la fase inicial del proyecto donde se dedica aproximadamente una semana, independientemente del formato, duración, etc, del resto de sprints. En este Sprint es donde se va a preparar el equipo tanto tecnológicamente como metodológicamente para que el desarrollo del proyecto tenga un buen comienzo, en especial en casos en los que se desconoce la metodología o no se tiene mucha práctica con ella.

El objetivo del Sprint 0 es preparar el conjunto del proyecto desde una perspectiva:

- Tecnológica.
- Metodológica.
- Organizativa para que el desarrollo del proyecto tenga un buen comienzo y mejor finalización.

Todo ello consiste básicamente en:

- **Definir con el cliente.** En primer lugar "the product owner", es decir, yo misma en este caso me estoy ocupando de definir con el cliente las características y funcionalidades del proyecto con el mayor número de detalles posibles. Con toda esa información construyo un documento en forma de historias de usuarios.

- **Construir el Product Backlog.** Las historias de usuario presentan unidades que pueden presentarse a los clientes como elementos acabados. Con ellas construimos el "Product Backlog".

- **Reuniones de equipo.** Se realiza una reunión con el equipo donde se presentan las historias de usuario y donde el equipo deberá: identificar lagunas de información, dificultades técnicas, dependencias entre historias, proponer creación/división de historias de usuario...

Los motivos para hacerlo de este modo, son que nos aporta las siguientes ventajas, siendo éstas de gran ayuda en el desarrollo del proyecto, mejorando la productividad del equipo.

- Obtenemos una definición contrastada con el cliente en forma de historias de usuarios.

- El equipo participa en la preparación del desarrollo identificando necesidades, dificultades, ventajas...

- La preparación de cada sprint será más fácil de realizar porque el equipo conoce con detalle el proyecto.

El Sprint 0 de este proyecto se podría identificar perfectamente con la fase de análisis descrita en capítulos anteriores de este documento.

Además de eso en el sprint se define la capacidad de trabajo del equipo por cada iteración. Para ello se han de definir el tamaño de la iteración y el rendimiento del equipo:

Tamaño del Sprint	2 Semanas (10 días laborables).
-------------------	---------------------------------

Trabajo por día	4 horas.
Horas por sprint	40 horas.

En este caso la cantidad de horas es relativamente baja para una iteración puesto que, aunque se simule la existencia de otros miembros en el equipo para estimaciones, reuniones, etc., debido a la naturaleza del proyecto el trabajo es realizado realmente por una única persona.

De esta forma para todo lo relacionado con el desarrollo en sí de la aplicación siempre se tendrá en consideración esta limitación. No siendo así para situaciones que puedan ser simuladas de forma sencilla y que den al proyecto un aspecto más real y natural [14].

Primer Sprint

Lo primero que se hace en la iteración es poner sobre la mesa el Product Backlog conseguido y utilizarlo para estimar. El mostrado aquí es una versión resumida de las historias de usuario presentadas en el capítulo de análisis.

Historia de Usuario	Miembro			Estimación media	Prioridad
	A	B	C		
HU01	20	20	40	26,66666667	Media
HU02	20	20	40	26,66666667	Media
HU03	13	13	20	15,33333333	Alta
HU04	13	13	20	15,33333333	Alta
HU05	13	13	20	15,33333333	Alta
HU06	13	13	20	15,33333333	Alta
HU07	5	3	8	5,33333333	Baja
HU08	3	3	5	3,66666667	Alta
HU09	13	8	13	11,33333333	Baja
HU10	20	20	13	17,66666667	Alta

Como se puede apreciar en la tabla varias historias tienen la misma prioridad, alta en este caso, por lo que al presentarse esta situación queda a juicio del equipo seleccionar la más prioritaria de entre ellas.

Horas máximas	40
Horas utilizadas	39
Horas restantes	1
Historia	Horas estim.
HU08	3,666666667
HU10	17,66666667
HU03	17,66666667

En este caso se observa que las historias de usuario que han sido seleccionadas para esta iteración son las historias:

Historia de Usuario	
ID	HU08
Nombre	Funcionamiento en Windows
Prioridad	Alta
Riesgo	Alto
Descripción	Como usuario quiero que el sistema funcione en el sistema operativo Windows 8.1
Validación	<ul style="list-style-type: none"> Quiero que la aplicación funcione en Windows 8.1 Diseño de la arquitectura de la aplicación

Historia de Usuario	
ID	HU10

Nombre	Base de datos
Prioridad	Alta
Riesgo	Alto
Descripción	Como desarrollador quiero que los datos introducidos sean persistentes.
Validación	<ul style="list-style-type: none"> - Quiero el uso de una base de datos. - Quiero que a la base de datos se le puedan introducir nuevos datos. - Quiero que los datos de la base de datos se puedan modificar. - Quiero que los datos de la base de datos se puedan eliminar.

Historia de Usuario	
ID	HU03
Nombre	Registrar y administrar artículos.
Prioridad	Alta
Riesgo	Alto
Descripción	Como gestor de compras quiero registrar los artículos nuevos en el sistema.
Validación	<ul style="list-style-type: none"> - Quiero poner introducir nuevos artículos en el sistema. - Los artículos deben agruparse por su tipología. - Cada artículo tiene asociado un precio de venta. - Los datos almacenados se pueden modificar posteriormente.

En primer lugar lo que se ha realizado es la historia de usuario número ocho, ya que en parte se encuentra hecha en el análisis. En cualquier caso no hay que confundir la parte del análisis con la que hay que hacer en el desarrollo. En el análisis se eligió una arquitectura y una configuración en base a esta historia de usuario, pero en este apartado lo que se va a hacer es poner en práctica los puntos establecidos anteriormente.

Se crea el proyecto en Visual Studio 2013, y se le nombra tfg01. A partir de ahí se estructura en capas siguiendo el patrón modelo – vista – controlador.

Se genera la solución tfg01, y dentro de este se crean los proyectos diferenciando las capas: Datos, Negocio y Vista.

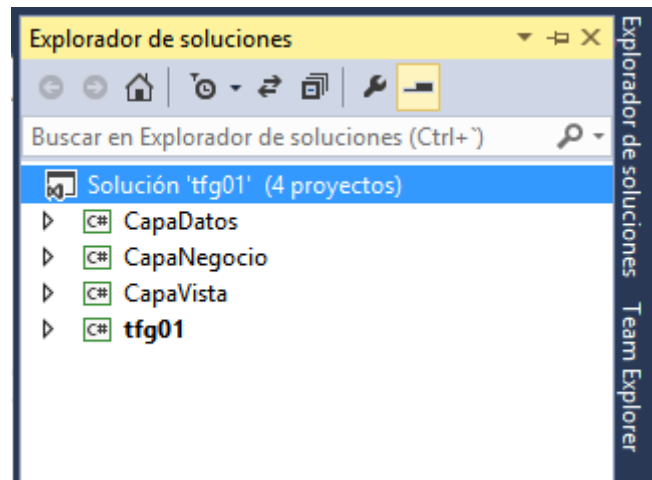


Ilustración 9 - Capas desarrollo

Después, para la historia de usuario 10 (HU10), que se centra en la creación de la base de datos, se genera una base de datos en el SGBD SQL Server:

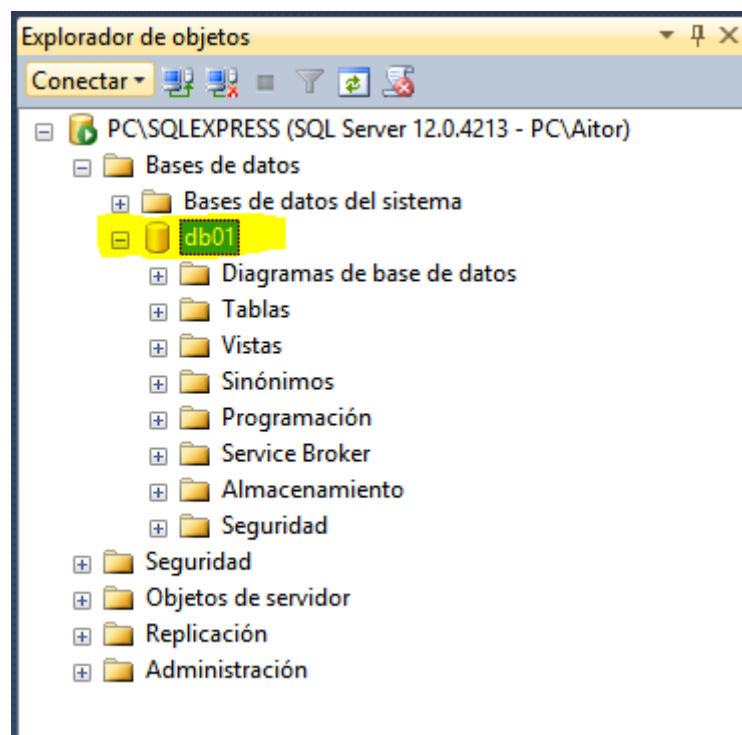


Ilustración 10 - Base de datos

Para la creación de la base de datos lo primero es crear un servidor dentro del SGBD, en este caso se crea PC.

Dentro de PC, en las bases de datos del sistema se crea la que se utilizará para este proyecto en concreto: db01.

Se trata de una base de datos relacional clásica.

De esta manera quedaría creada la base de datos para el desarrollo del sistema. Estando disponible en todo momento la posibilidad de añadir tablas o datos, así como eliminarlas o modificarlas en cualquier momento.

En este momento ya se tiene preparada la base de datos vacía en nuestro SGBD y es hora de diseñar esta misma. Para ello se determina la necesidad de las siguientes tablas:

- Artículo.
- Centro.
- Cliente.
- Compra.
- Empleado.
- Grupo de artículos.
- Proveedor.
- Stock.
- Venta.

Para el diseño completo se diferenciarán tres fases que ayudarán en el momento del desarrollo por sprints, teniendo así en cada iteración un esquema de las tablas a utilizar.

Primera fase:

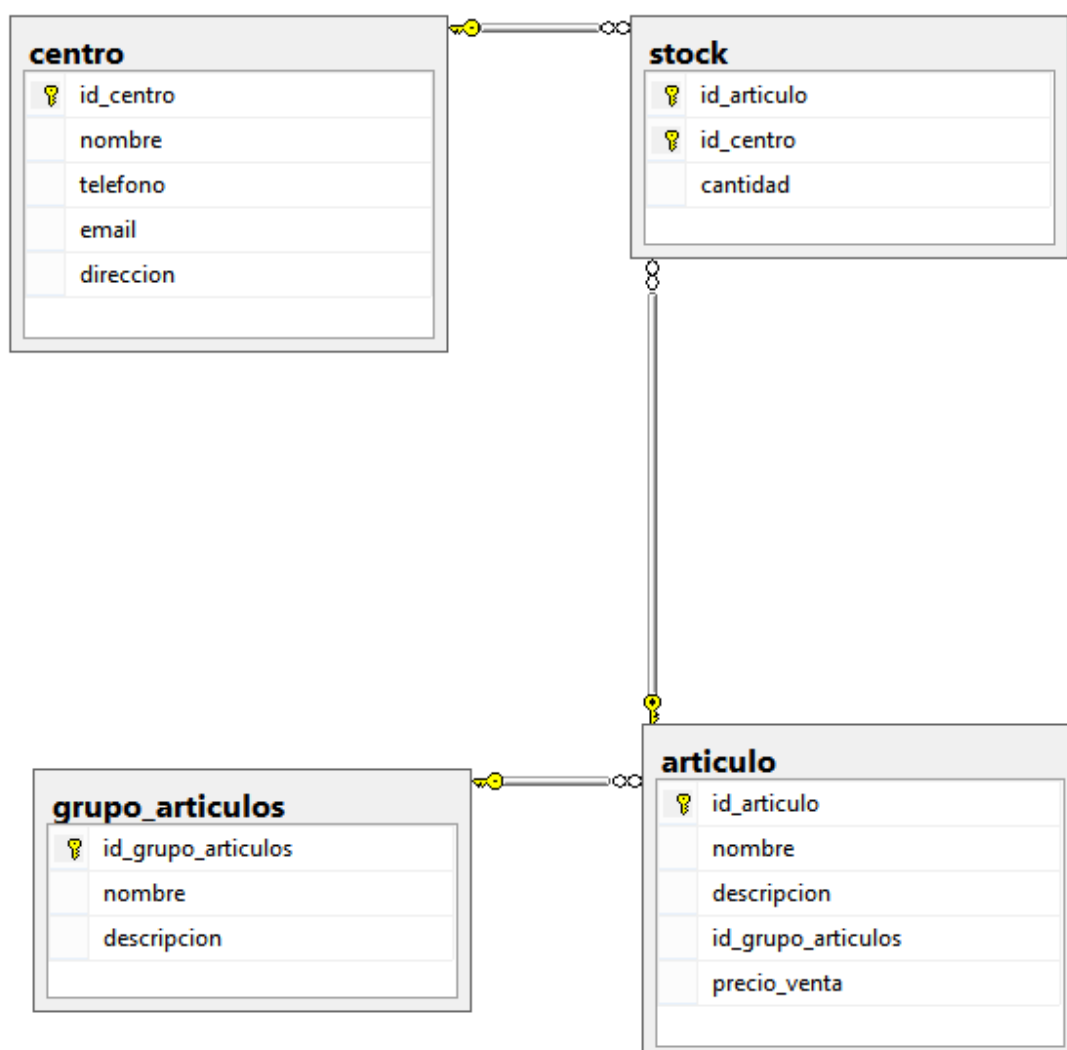


Ilustración 11 - Modelo base datos I

Segunda fase:

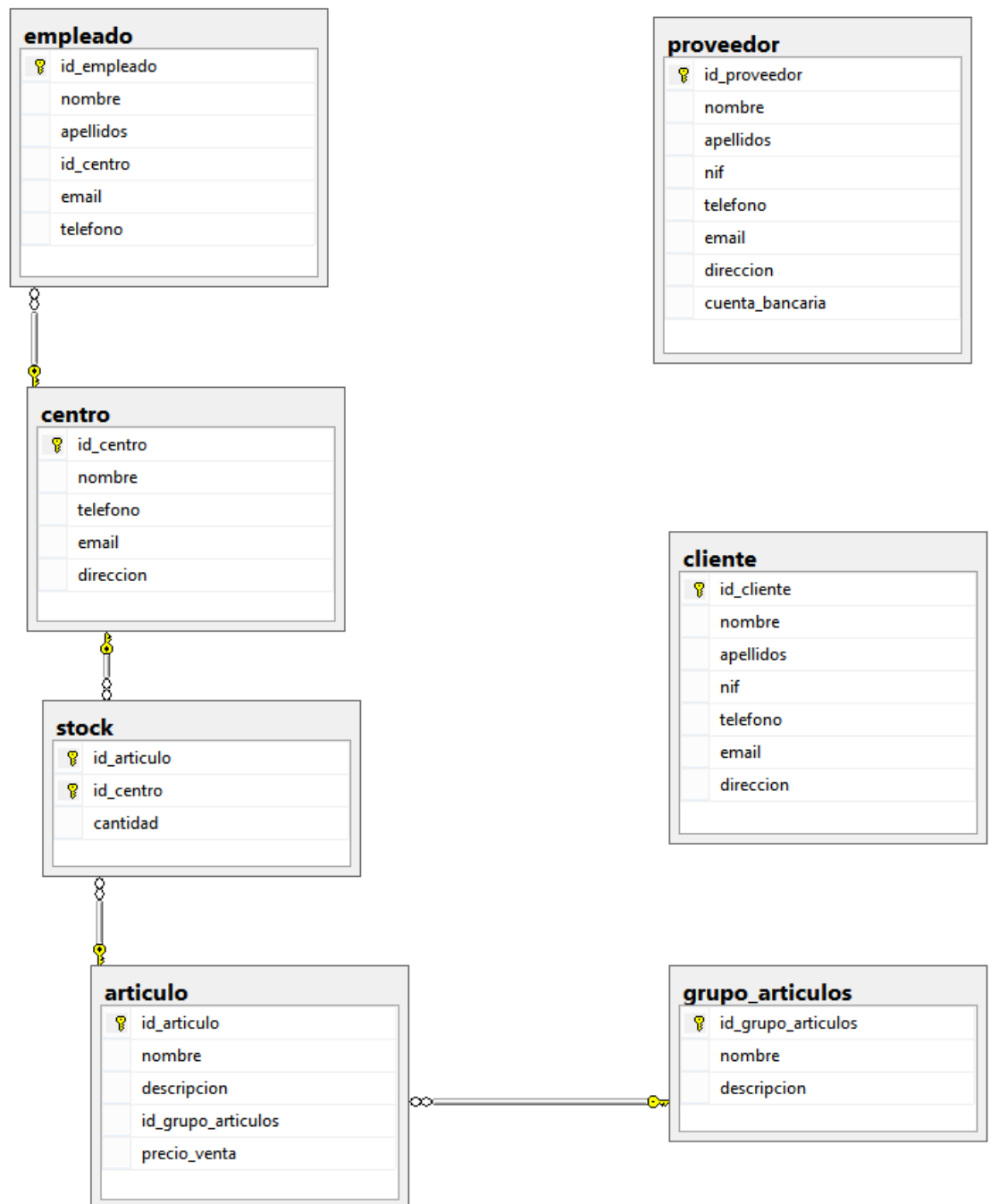


Ilustración 12 - Modelo base de datos II

Tercera fase:

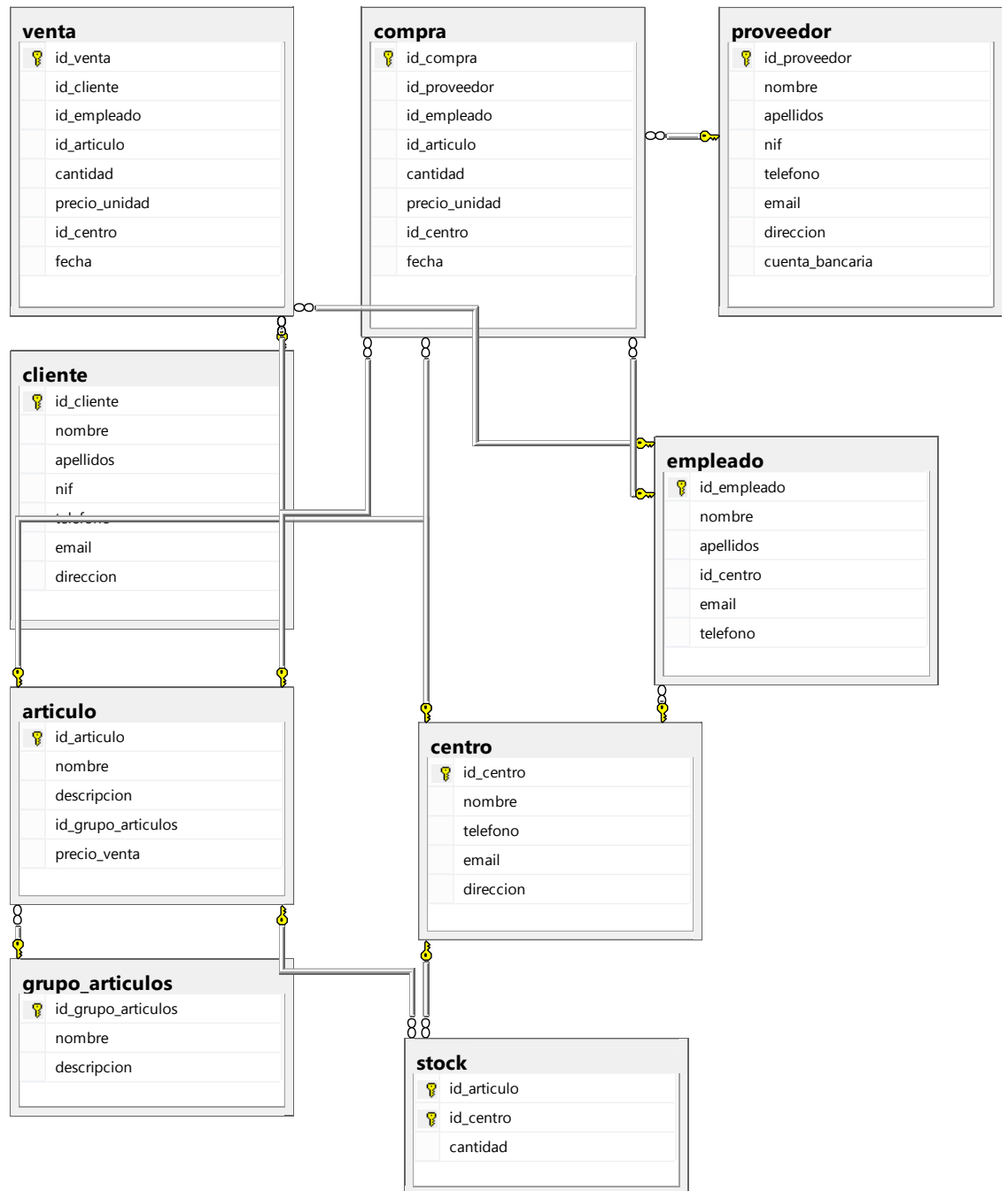


Ilustración 13 - Modelo base de datos III

Con este proceso el diseño de la base de datos quedaría listo y se podría pasar al siguiente punto del desarrollo.

El siguiente paso será abordar la HU03. En esta historia de usuario se pide la gestión de artículos. Para ello será necesario abordar la gestión de los grupos de artículos ya que los artículos tienen dependencia de esto.

El primer paso será la creación de los procedimientos almacenados:

- Buscar en grupo_artículos:

```
USE [db01]
GO
/***** Object: StoredProcedure [dbo].[spbuscar_grupo_articulos]    Script Date: 09/09/2015 20:56:01 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
ALTER proc [dbo].[spbuscar_grupo_articulos]
@textobuscar varchar(50)
as
select * from grupo_articulos
where nombre like @textobuscar + '%' --Alt +39
|
```

Ilustración 14 - Procedimiento buscar

- Insertar en grupo_artículos:

```
USE [db01]
GO
/***** Object: StoredProcedure [dbo].[spinsertar_grupo_articulos]    Script Date: 09/09/2015 21:02:28 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
ALTER proc [dbo].[spinsertar_grupo_articulos]
@id_grupo_articulos int output,
@nombre varchar(50),
@descripcion varchar (150)
as
insert into grupo_articulos (nombre, descripcion)
values (@nombre, @descripcion)
```

Ilustración 15 - Procedimiento insertar

- Eliminar de grupo_artículos:

```

USE [db01]
GO
/***** Object: StoredProcedure [dbo].[speliminar_grupo_articulos]    Script Date: 09/09/2015 21:01:53 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
ALTER proc [dbo].[speliminar_grupo_articulos]
@id_grupo_articulos int
as
delete from grupo_articulos
where id_grupo_articulos=@id_grupo_articulos

```

Ilustración 16 - Procedimiento eliminar

- Mostar contenido de grupo_artículos:

```

USE [db01]
GO
/***** Object: StoredProcedure [dbo].[spmostrar_grupo_articulos]    Script Date: 09/09/2015 21:02:54 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
ALTER proc [dbo].[spmostrar_grupo_articulos]
as
select top 220 * from grupo_articulos
order by id_grupo_articulos desc

```

Ilustración 17 - Procedimiento mostrar

- Editar elemento de grupo_artículos:

```

USE [db01]
GO
/***** Object: StoredProcedure [dbo].[speditar_grupo_articulos]    Script Date: 09/09/2015 21:01:09 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
ALTER proc [dbo].[speditar_grupo_articulos]
@id_grupo_articulos int,
@nombre varchar(50),
@descripcion varchar (150)
as
update grupo_articulos set nombre=@nombre,
descripcion=@descripcion
where id_grupo_articulos=@id_grupo_articulos

```

Ilustración 18 - Procedimiento editar

Con estos procedimientos almacenados quedaría completa la gestión (consulta, inserción, eliminación o modificación) de los registros de la tabla de grupo_artículos en la parte del SGBD.

El siguiente paso es crear la conexión a la base de datos desde el programa cliente. Para ello se abre el Visual Studio 2013 se crea una nueva solución de Windows Forms, se crean las tres capas anteriormente cometnadas dentro de la solución a modo de proyectos independientes, se les asignan las dependencias entre ellos siguiendo las relaciones mencionadas en el apartado

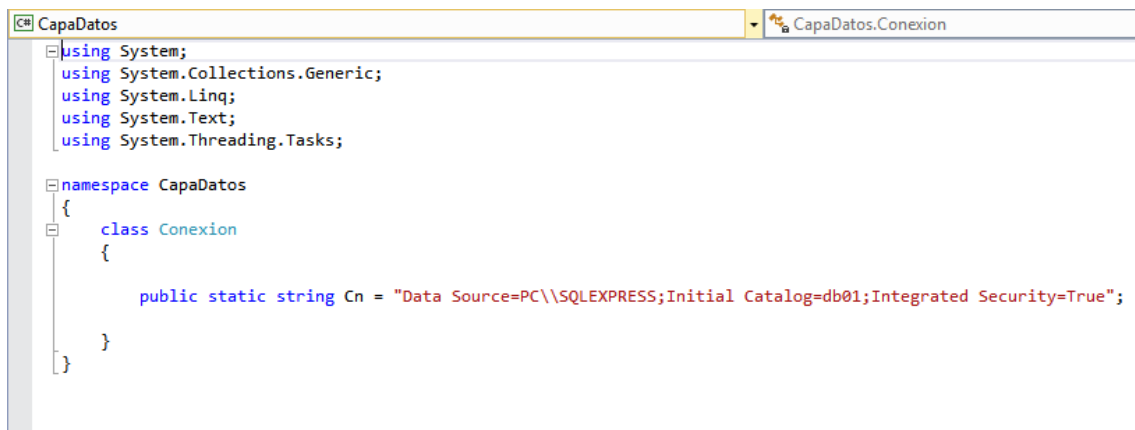


Ilustración 19 - Conexión base de datos

MCV y se crea la clase conexión dentro de la capa de datos del programa:

Siendo Cn la conexión a la base de datos alojada en el servidor especificado en la ruta.

En este momento ya se dispone de una base de datos y la conexión a esta desde la aplicación. El siguiente paso a dar es la creación en la capa de datos de la clase DGrupo_articulos, que se encargará de ser la interfaz entre la base de datos y el resto de la aplicación.

Este esta clase resumida resultante:

```
namespace CapaDatos
{
    public class DGrupo_articulos
    {
        private int _Id_grupo_articulos;
        private string _Nombre;
        private string _Descripcion;
    }
}
```

```

private string _TextoBuscar;

//Insertar
public string Insertar(DGrupo_articulos Grupo)
{
    Introduce los datos en los parámetros del procedimiento de la
    base de datos y hace la llamada a este para su ejecución e insertar los
    datos.

}

//Editar
public string Editar(DGrupo_articulos Grupo)
{
    Introduce los datos en los parámetros del procedimiento de la
    base de datos y hace la llamada a este para su ejecución y editar así el
    registro indicado.

}

//Eliminar
public string Eliminar(DGrupo_articulos Grupo)
{
    Introduce los datos en los parámetros del procedimiento de la base
    de datos y hace la llamada a este para su ejecución y eliminar el registro
    indicado.

}

//Mostrar
public DataTable Mostrar()
{
    Devuelve todos los registros de la tabla.

}

//Buscar por nombre
public DataTable BuscarNombre(DGrupo_articulos Grupo)
{
    Devuelve el registro con el nombre indicado.

}

```

Una vez ya se dispone de la interfaz con la base de datos el siguiente paso es el desarrollo de la capa negocio. En la cual se incluye la clase NGrupo_articulos. Esta clase sirve de interfaz entre la capa de datos y la interfaz

de usuario alojada en la capa vista. Además se encarga de la lógica de la aplicación y de realizar cualquier operación que fuese necesaria.

```
namespace CapaNegocio
{
    public class NGrupo_articulos
    {
        //Método Insertar -> llama al insertar de DGrupo_articulos
        public static string Insertar (string nombre, string descripcion)

        //Método Editar -> llama al editar de DGrupo_articulos
        public static string Editar(int idgrupo_articulos, string nombre, string
descripcion)

        //Método Eliminar -> llama al eliminar de DGrupo_articulos
        public static string Eliminar(int idgrupo_articulos)

        //Método Mostrar -> llama al mostrar de DGrupo_articulos
        public static DataTable Mostrar()

        //Método Buscarnombre -> llama al BuscarNombre de DGrupo_articulos
        public static DataTable BuscarNombre(string nombre)
    }
}
```

Por último se ha de desarrollar la interfaz gráfica para que el usuario pueda administrar e interactuar con la aplicación de forma amigable. Para ello se genera la capa vista y se crea la clase Frm_Grupo_articulos.

El diseño de la interfaz de usuario quedaría de la siguiente manera:

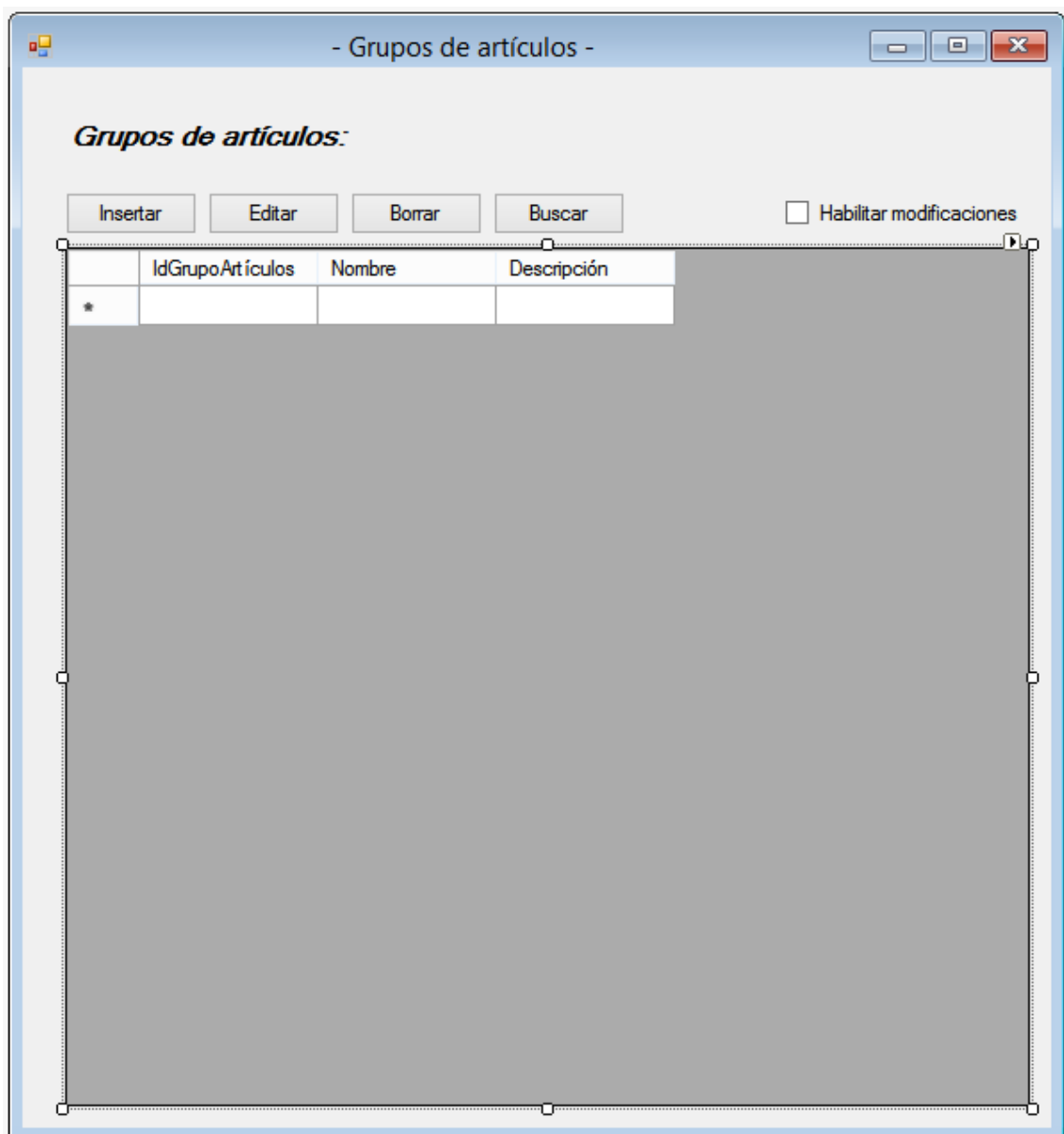


Ilustración 20 - Interfaz de usuario Grupo de artículos

Finalizadas las tareas para administrar los grupos de artículos se procede a realizar los pasos equivalentes con la tabla artículos. Los pasos a dar son similares a los mostrados con grupo de artículos.

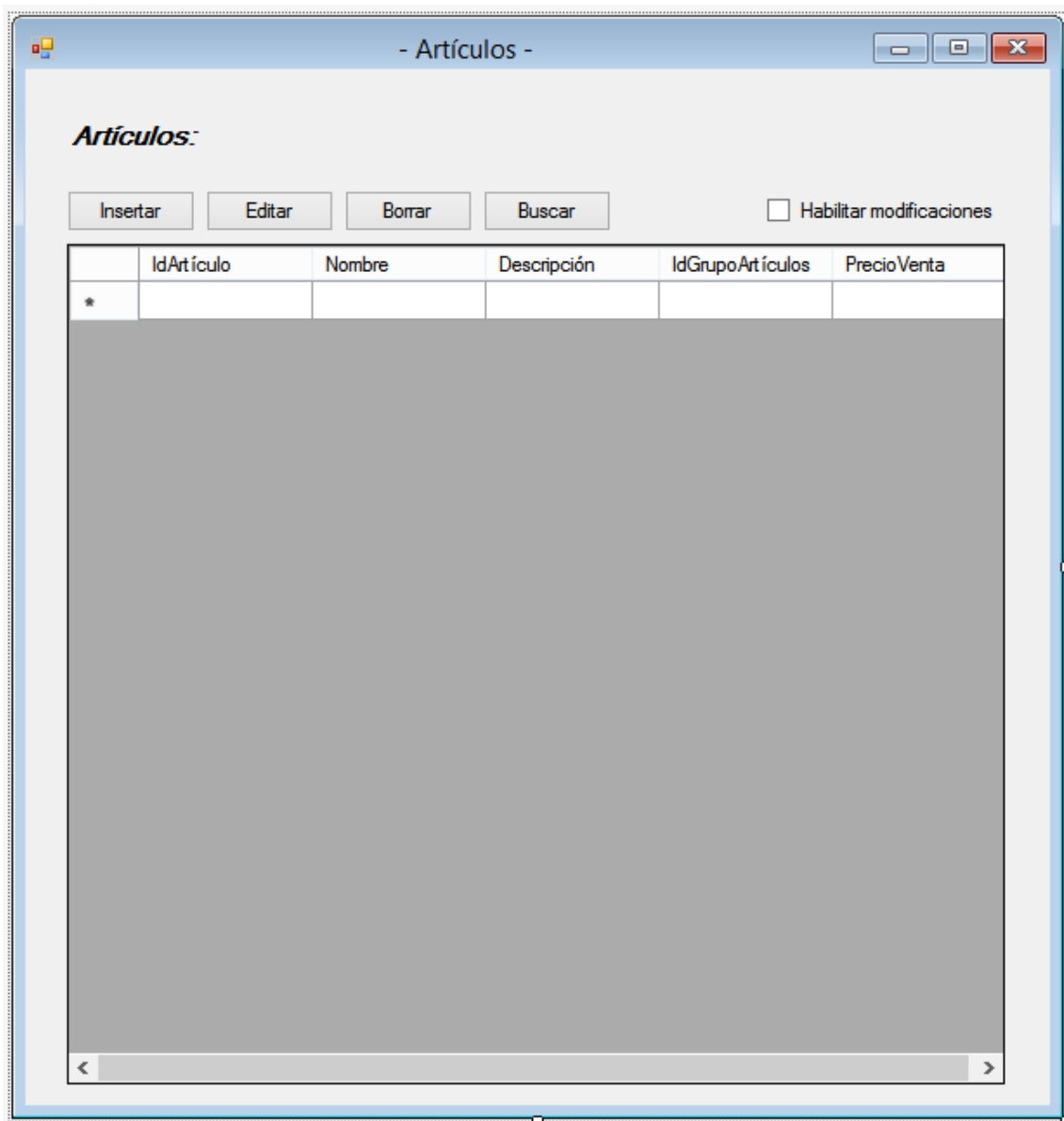


Ilustración 21 - Interfaz de usuario Artículos

Con esto finalizarían las tareas del primer sprint. Como resumen del Sprint 1 se podría decir que se ha dejado la base para continuar el desarrollo de la aplicación añadiendo todas las funcionalidades necesarias al sistema para completar las historias de usuario establecidas.

Tarea	Historia de Usuario	Estado inicial	Estado estimado	Estado final
T01	HU03	No iniciada	Completada	Completada
T02	HU03	No iniciada	Completada	Completada
T03	HU03	No iniciada	Completada	Completada
T04	HU03	No iniciada	Completada	Completada
T05	HU03	No iniciada	Completada	Completada
T06	HU03	No iniciada	Completada	Completada
T07	HU03	No iniciada	Completada	Completada
T08	HU03	No iniciada	Completada	Completada
T09	HU03	No iniciada	Completada	Completada
T10	HU03	No iniciada	Completada	Completada
T11	HU05	No iniciada	No iniciada	No iniciada
T12	HU05	No iniciada	No iniciada	No iniciada
T13	HU05	No iniciada	No iniciada	No iniciada
T14	HU05	No iniciada	No iniciada	No iniciada
T15	HU05	No iniciada	No iniciada	No iniciada
T16	HU06	No iniciada	No iniciada	No iniciada
T17	HU06	No iniciada	No iniciada	No iniciada
T18	HU06	No iniciada	No iniciada	No iniciada
T19	HU06	No iniciada	No iniciada	No iniciada
T20	HU06	No iniciada	No iniciada	No iniciada
T21	HU06	No iniciada	No iniciada	No iniciada
T22	HU07	No iniciada	No iniciada	No iniciada
T23	HU10	No iniciada	Completada	Completada
T24	HU10	No iniciada	Completada	Completada
T25	HU04	No iniciada	No iniciada	No iniciada
T26	HU04	No iniciada	No iniciada	No iniciada
T27	HU04	No iniciada	No iniciada	No iniciada
T28	HU06	No iniciada	No iniciada	No iniciada
T29	HU06	No iniciada	No iniciada	No iniciada
T30	HU01	No iniciada	No iniciada	No iniciada
T31	HU01	No iniciada	No iniciada	No iniciada
T32	HU01	No iniciada	No iniciada	No iniciada
T33	HU02	No iniciada	No iniciada	No iniciada
T34	HU02	No iniciada	No iniciada	No iniciada
T35	HU02	No iniciada	No iniciada	No iniciada
T36	HU02	No iniciada	No iniciada	No iniciada
T37	HU09	No iniciada	No iniciada	No iniciada
T38	HU09	No iniciada	No iniciada	No iniciada
T39	HU09	No iniciada	No iniciada	No iniciada
T40	HU08	No iniciada	Completada	Completada

Finalmente el esfuerzo dedicado al primer sprint se recoge en la siguiente tabla:

Historia de usuario	Tiempo estimado	Tiempo final
HU08	3,666666667	3
HU10	17,66666667	16,5
HU03	17,66666667	20
Total	39	39,5

- Pruebas funcionales:

Id	P0101
Resultado	Válido
Sistema	Cliente
Descripción	Comprobación que el cliente funciona en Windows.
Acciones	1. Iniciar la aplicación en el sistema operativo Windows.

Id	P0102
Resultado	Válido
Sistema	Base de datos
Descripción	Mediante el SGBD insertar un registro en la tabla grupo de artículos.
Acciones	1. Iniciar el SGBD. 2. Introducir un registro en la tabla grupo_artículos. 3. Realizar una consulta del contenido de la tabla grupo_artículos.

Id	P0103
Resultado	Válido
Sistema	Base de datos
Descripción	Mediante el SGBD insertar un registro en la tabla artículos.

Acciones	<ol style="list-style-type: none"> 1. Iniciar el SGBD. 2. Introducir un registro en la tabla artículos. 3. Realizar una consulta del contenido de la tabla artículos.
----------	--

Id	P0104
Resultado	Válido
Sistema	Cliente / Base de datos
Descripción	Comprobación que la conexión entre ambos sistemas funciona correctamente.
Acciones	<ol style="list-style-type: none"> 1. Iniciar el SGBD. 2. Insertar un registro en la tabla artículos. 3. Iniciar la aplicación cliente. 4. Abrir la ventana de artículos. 5. Realizar una consulta del contenido de la tabla artículos.

Segundo Sprint

En este segundo Sprint se afrontarán las tareas relacionadas con los las siguientes historias de usuario en orden de prioridad. Además se reestiman los esfuerzos necesarios en base a la experiencia adquirida durante el primer Sprint:

Historia de Usuario	Miembro			Estimación media	Prioridad
	A	B	C		
HU01	20	20	40	26,66666667	Media
HU02	20	20	40	26,66666667	Media
HU03					Alta
HU04	13	8	13	11,33333333	Alta
HU05	13	8	13	11,33333333	Alta
HU06	13	13	20	15,33333333	Alta

HU07	5	3	8	5,33333333	Baja
HU08					Alta
HU09	13	8	13	11,33333333	Baja
HU10					Alta

Tal como se observa en la tabla las historias de usuario que se adjudican a esta iteración son las siguientes:

Horas máximas	40
Horas utilizadas	38
Horas restantes	2
Historia	Horas estim.
HU04	11,33333
HU05	11,33333
HU06	15,33333

Los detalles de las historias de usuario seleccionadas se muestran en estas tablas:

Historia de Usuario	
ID	HU04
Nombre	Registrar y administrar proveedores.
Prioridad	Alta
Riesgo	Alto
Descripción	Como gestor de compras quiero registrar nuevos proveedores.
Validación	<ul style="list-style-type: none"> - Quiero poder introducir nuevos proveedores en el sistema. - Cada proveedor debe tener relacionado un nombre, un NIF, un email y una cuenta bancaria. - Puede haber almacenados dos proveedores con el mismo NIF.

	- Los datos almacenados se pueden modificar posteriormente.
--	---

Historia de Usuario	
ID	HU05
Nombre	Registrar y administrar centros.
Prioridad	Alta
Riesgo	Alto
Descripción	Como gestor de almacenes quiero disponer de diferentes centros como destino / origen de las mercancías.
Validación	<ul style="list-style-type: none"> - Quiero poder introducir nuevos centros en el sistema. - Cada centro debe tener relacionado un nombre, un teléfono, un email y una dirección física. - Los datos almacenados se pueden modificar posteriormente.

Historia de Usuario	
ID	HU06
Nombre	Control de stocks.
Prioridad	Alta
Riesgo	Medio
Descripción	Como gestor de almacenes quiero tener control de los stocks de los artículos que hay en los centros.
Validación	<ul style="list-style-type: none"> - Quiero incluir nuevos artículos a un centro. - Quiero aumentar o reducir la cantidad almacenada. - Quiero poder tener el mismo artículo en varios centros.

El desarrollo de este sprint debido a que las tareas a realizar son de una ejecución similar a la desarrollada y explicada en el primer sprint con la implementación de los grupos de artículos, no se va a desarrollar al completo sino que se hará una simulación.

A continuación se muestran las interfaces de usuario de cada una de las historias de usuario:

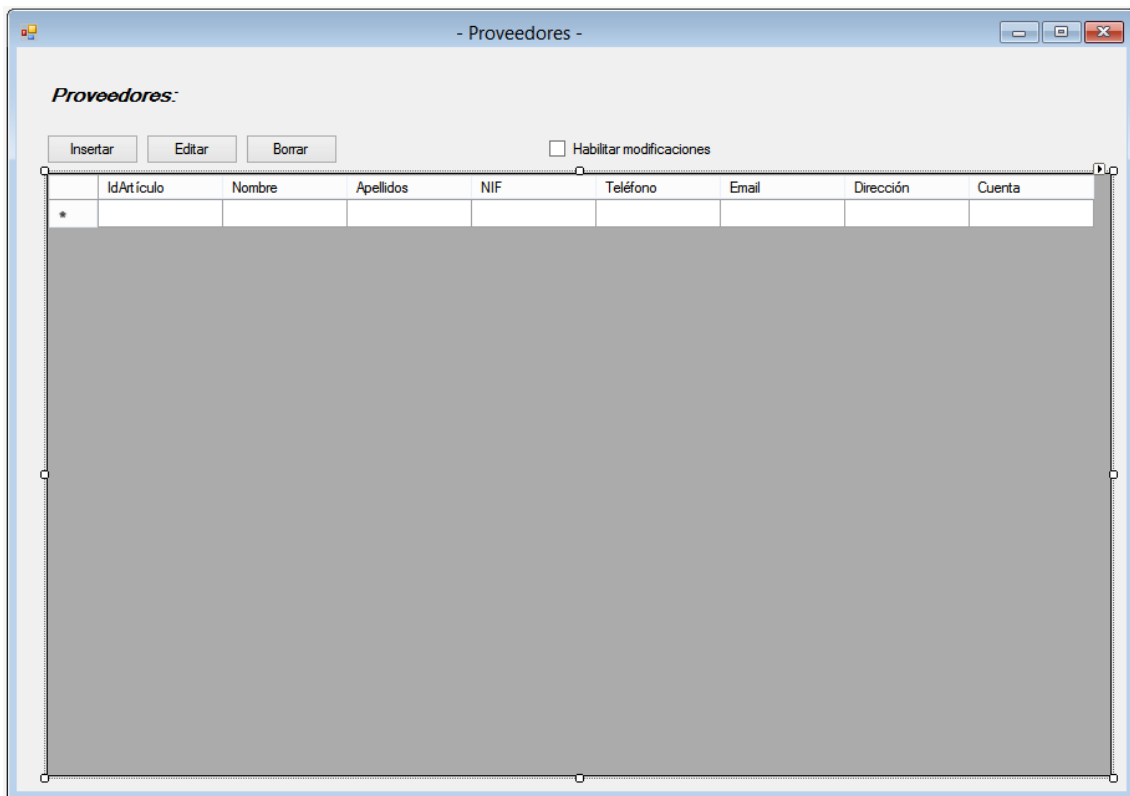


Ilustración 22 - Interfaz de usuario Proveedores

En esta imagen se muestra la interfaz para la gestión de proveedores, que sigue la misma línea de diseño que la anteriormente mostrada de artículos.

Después se repite el proceso con la historia de usuario 5, en la cual se han de hacer los desarrollos equivalentes pero partiendo de la tabla centros de la base de datos:

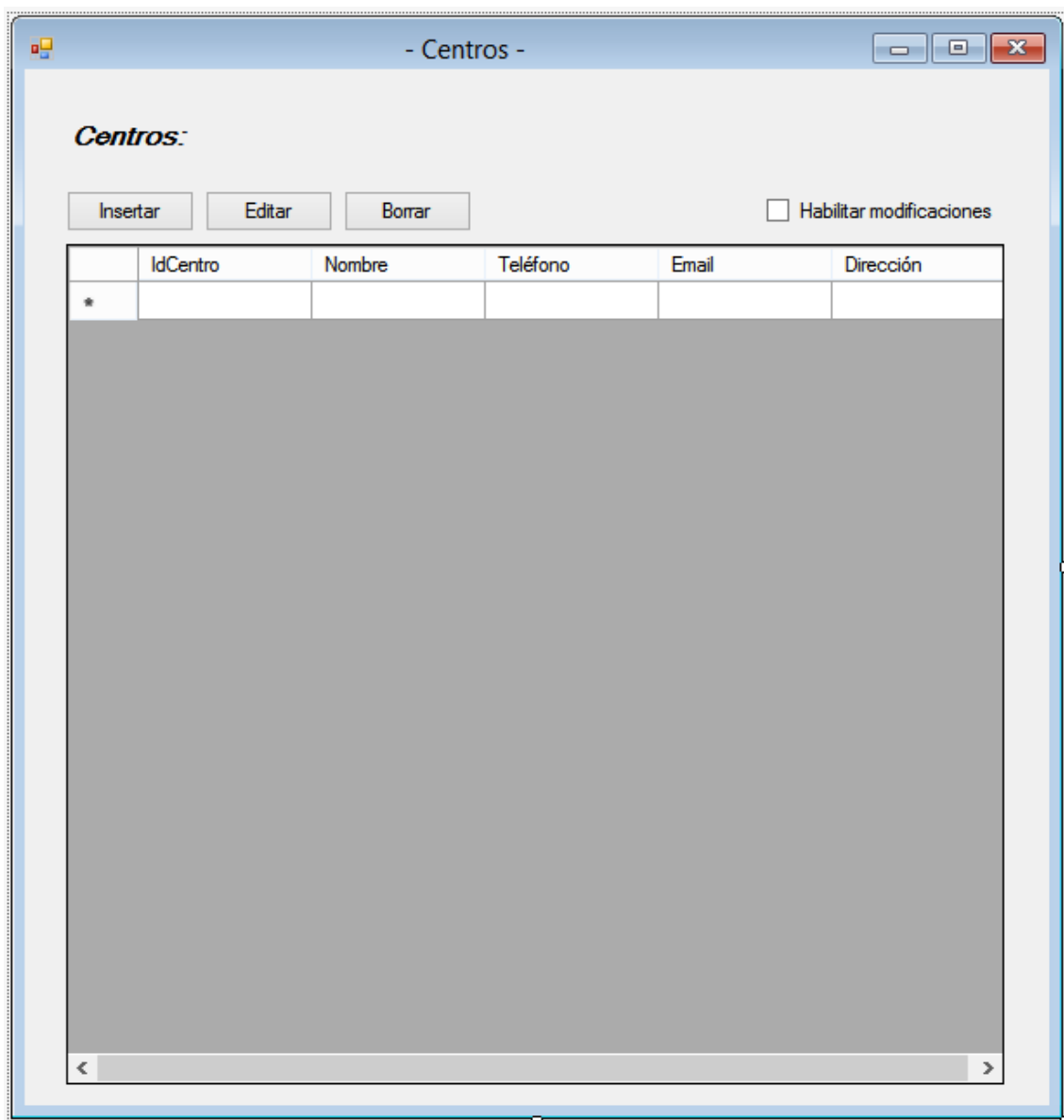


Ilustración 23 - Interfaz de usuario Centros

Finalmente lo último a realizar en esta iteración será la historia de usuario 6, en la cual hay que crear lo necesario para administrar los stocks. En parte este desarrollo es muy similar a los anteriores, especialmente en la capa de datos. Pero en la capa negocio se debe controlar los stocks, es decir, no permitir que se queden en negativo ya que eso no sería posible.

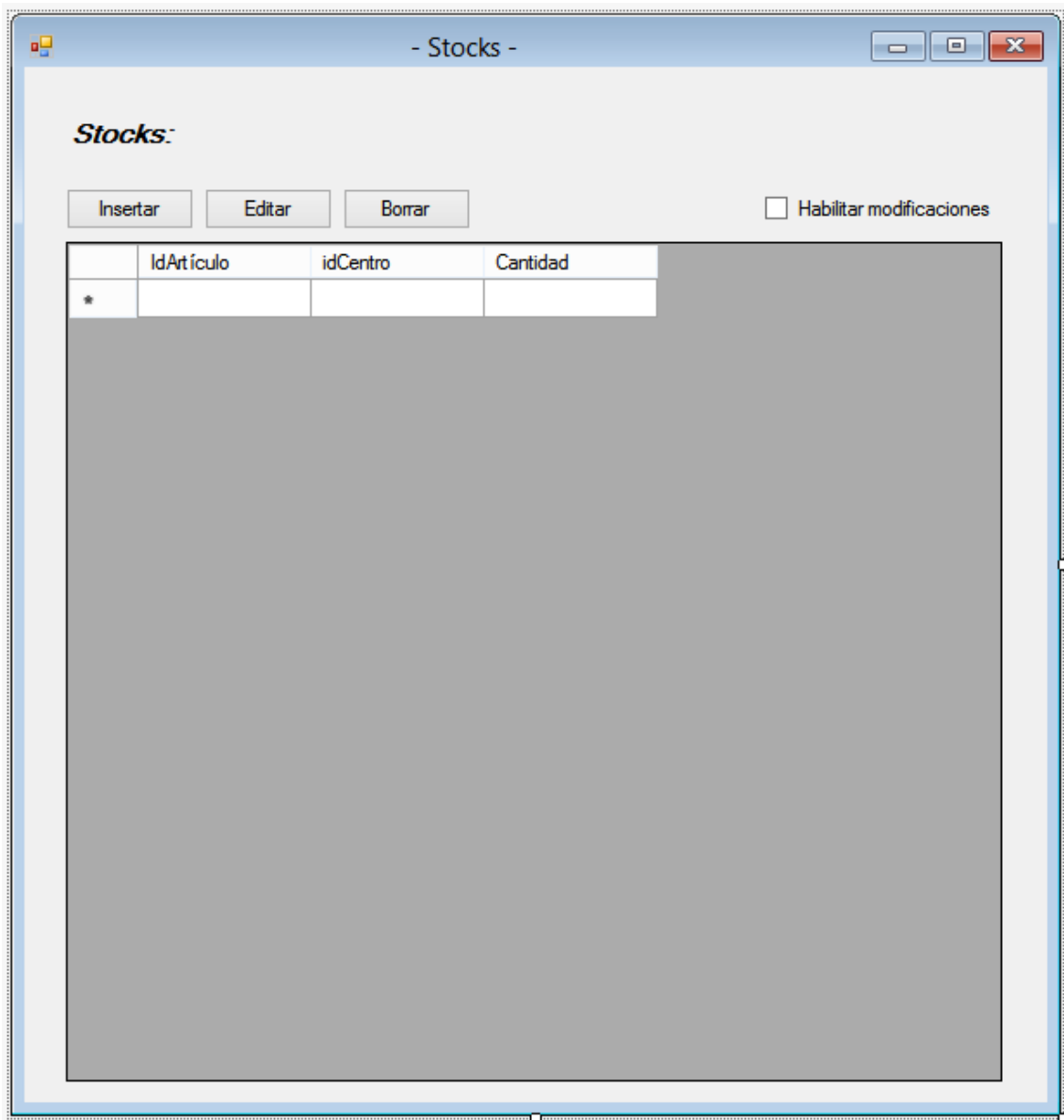


Ilustración 24 - Interfaz de usuario Stocks

Por último se muestra la interfaz de usuario para administrar los stocks directamente:

El resumen de las tareas relacionadas con el Sprint 2 quedaría de la siguiente forma:

Tarea	Historia de Usuario	Estado inicial	Estado estimado	Estado final
T01	HU03	Completada	Completada	Completada
T02	HU03	Completada	Completada	Completada
T03	HU03	Completada	Completada	Completada
T04	HU03	Completada	Completada	Completada
T05	HU03	Completada	Completada	Completada
T06	HU03	Completada	Completada	Completada
T07	HU03	Completada	Completada	Completada
T08	HU03	Completada	Completada	Completada
T09	HU03	Completada	Completada	Completada
T10	HU03	Completada	Completada	Completada
T11	HU05	Completada	Completada	Completada
T12	HU05	Completada	Completada	Completada
T13	HU05	Completada	Completada	Completada
T14	HU05	Completada	Completada	Completada
T15	HU05	Completada	Completada	Completada
T16	HU06	Completada	Completada	Completada
T17	HU06	Completada	Completada	Completada
T18	HU06	Completada	Completada	Completada
T19	HU06	Completada	Completada	Completada
T20	HU06	Completada	Completada	Completada
T21	HU06	Completada	Completada	Completada
T22	HU07	No iniciada	No iniciada	No iniciada
T23	HU10	Completada	Completada	Completada
T24	HU10	Completada	Completada	Completada
T25	HU04	Completada	Completada	Completada
T26	HU04	Completada	Completada	Completada
T27	HU04	Completada	Completada	Completada
T28	HU06	Completada	Completada	Completada
T29	HU06	Completada	Completada	Completada
T30	HU01	No iniciada	No iniciada	No iniciada
T31	HU01	No iniciada	No iniciada	No iniciada
T32	HU01	No iniciada	No iniciada	No iniciada
T33	HU02	No iniciada	No iniciada	No iniciada
T34	HU02	No iniciada	No iniciada	No iniciada
T35	HU02	No iniciada	No iniciada	No iniciada
T36	HU02	No iniciada	No iniciada	No iniciada
T37	HU09	No iniciada	No iniciada	No iniciada
T38	HU09	No iniciada	No iniciada	No iniciada

T39	HU09	No iniciada	No iniciada	No iniciada
T40	HU08	Completada	Completada	Completada
T41	HU01	No iniciada	No iniciada	No iniciada
T42	HU01	No iniciada	No iniciada	No iniciada
T43	HU01	No iniciada	No iniciada	No iniciada
T44	HU01	No iniciada	No iniciada	No iniciada
T45	HU01	No iniciada	No iniciada	No iniciada
T46	HU01	No iniciada	No iniciada	No iniciada
T47	HU01	No iniciada	No iniciada	No iniciada
T48	HU01	No iniciada	No iniciada	No iniciada
T49	HU01	No iniciada	No iniciada	No iniciada
T50	HU01	No iniciada	No iniciada	No iniciada

Resumen de esfuerzo dedicado a cada historia de usuario desarrollada:

Historia de usuario	Tiempo estimado	Tiempo final
HU04	11,33333	11
HU05	11,33333	10
HU06	15,33333	16
Total	38	37

- Pruebas funcionales:

Id	P0201
Resultado	Válido
Sistema	Base de datos
Descripción	Mediante el SGBD insertar un registro en la tabla proveedores.
Acciones	4. Iniciar el SGBD. 5. Introducir un registro en la tabla proveedores. 6. Realizar una consulta del contenido de la tabla proveedores.

Id	P0202
Resultado	Válido
Sistema	Base de datos

Descripción	Mediante el SGBD insertar un registro en la tabla centros.
Acciones	<ol style="list-style-type: none"> 7. Iniciar el SGBD. 8. Introducir un registro en la tabla centros. 9. Realizar una consulta del contenido de la tabla centros.

Id	P0203
Resultado	Válido
Sistema	Base de datos
Descripción	Comprobar que un artículo se puede encontrar en diferentes centros.
Acciones	<ol style="list-style-type: none"> 1. Iniciar el SGBD 2. Introducir un registro en la tabla stocks con centro = 1. 3. Introducir otro registro usando un artículo diferente usando centro = 2.

Id	P0204
Resultado	Válido
Sistema	Cliente / Base de datos
Descripción	Comprobar que se insertan correctamente registros desde las interfaces de usuario.
Acciones	<ol style="list-style-type: none"> 1. Iniciar SGBD 2. Iniciar cliente -> Centros. 3. Insertar un registro. 4. Realizar una consulta de la tabla

Tercer Sprint

Las siguientes historias de usuario a realizar en esta iteración son las siguientes después de reestimar y ordenarlas por prioridad:

Historia de Usuario	Miembro			Estimación media	Prioridad
	A	B	C		
HU01	40	40	40	40	Media
HU02	20	20	20	20	Media
HU03					Alta
HU04					Alta
HU05					Alta
HU06					Alta
HU07	5	3	8	5,33333333	Baja
HU08					Alta
HU09	13	8	13	11,33333333	Baja
HU10					Alta

Según la tabla de historias de usuario la siguiente a realizar serían las HU01 y HU02. Debido a que la HU02 tiene cierta dependencia de la HU01 se decide que esta será la que se lleve a cabo durante esta iteración:

Horas máximas	40
Horas utilizadas	40
Horas restantes	0
Historia	Horas estim.
HU01	40

Los detalles de esta historia de usuario son los siguientes:

Historia de Usuario	
ID	HU01
Nombre	Introducir ventas realizadas.
Prioridad	Media
Riesgo	Bajo
Descripción	Como gestor de ventas quiero introducir las operaciones de venta de artículos realizadas en el sistema.

Validación	<ul style="list-style-type: none"> - Quiero que la operación quede registrada a nombre de un empleado. - Quiero que la operación se diferencie por centro y cliente. - Quiero que cada venta de un tipo de artículo se haga por separado. - Quiero que en la venta se almacene el precio del artículo en ese momento.

En este sprint se tiene que desarrollar la funcionalidad de registrar ventas, es decir, registrar ventas realizadas por un empleado a un cliente. Estas ventas son de un artículo del cual se indica la cantidad vendida del producto.

Para poder realizar todas estas operaciones reclamadas en la historia de usuario 1 se deben crear las tablas de empleado, cliente y venta en la base de datos. Además dentro del SGBD también hay que crear los procedimientos necesarios para las modificaciones del contenido de las tablas.

Después hay que completar la capa de Datos en la aplicación. Se crean las clases DClientes.cs, DEmpleados.cs y DVentas.cs para interactuar con la base de datos. Estas clases tendrán los métodos correspondientes para hacer uso de los procedimientos almacenados en el SGBD para el manejo de los datos.

A continuación se crean en la capa de negocio las clases NClientes.cs, NEmpleados.cs y NVentas.cs. En este caso las clases de empleados y clientes van a funcionar de forma similar a lo desarrollado hasta el momento, en la capa negocio simplemente se conectan las capas de datos y vista de la misma tabla. Pero con la clase NVentas.cs el funcionamiento cambia ligeramente de una forma en la cual esta clase además de usar funciones de la clase correspondiente de ventas de la capa de datos, también utiliza funciones de otras tablas, ya que al realizar una venta se modifican los stocks disponibles.

Finalmente se desarrollan las clases de la capa vista que incluyen las interfaces de usuario para la administración de estas tres tablas.

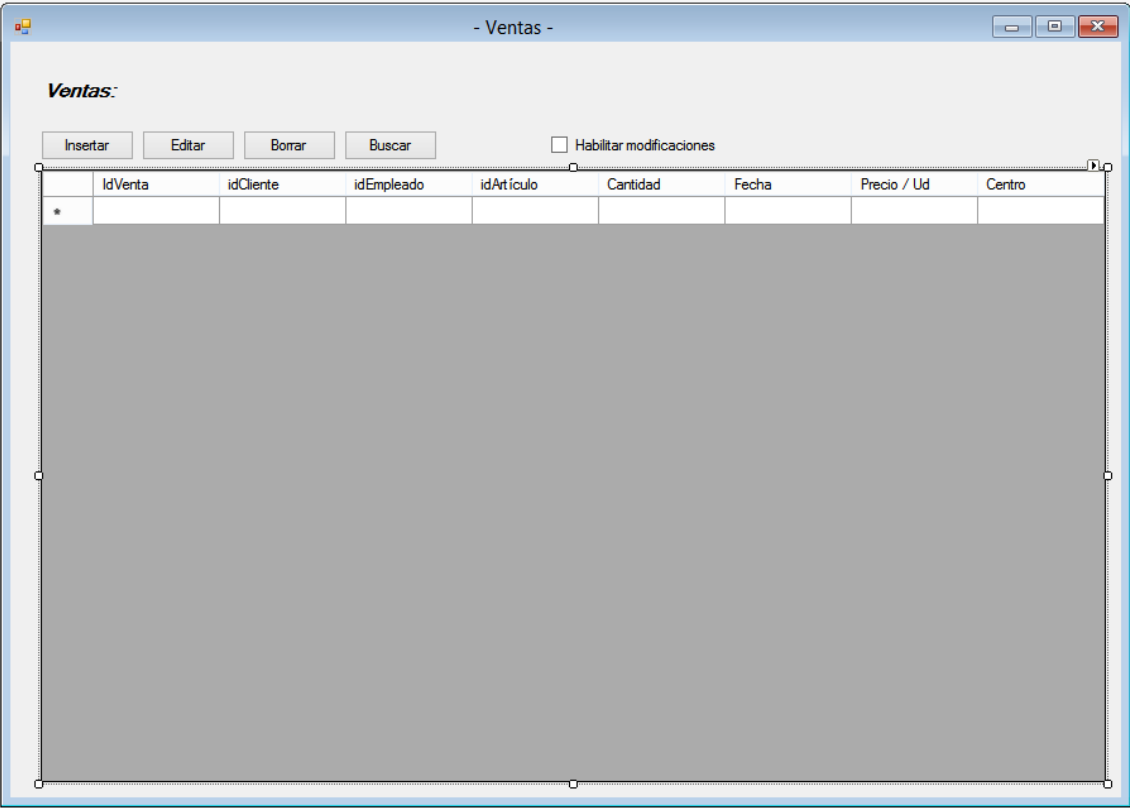


Ilustración 25 - Interfaz de usuario Ventas

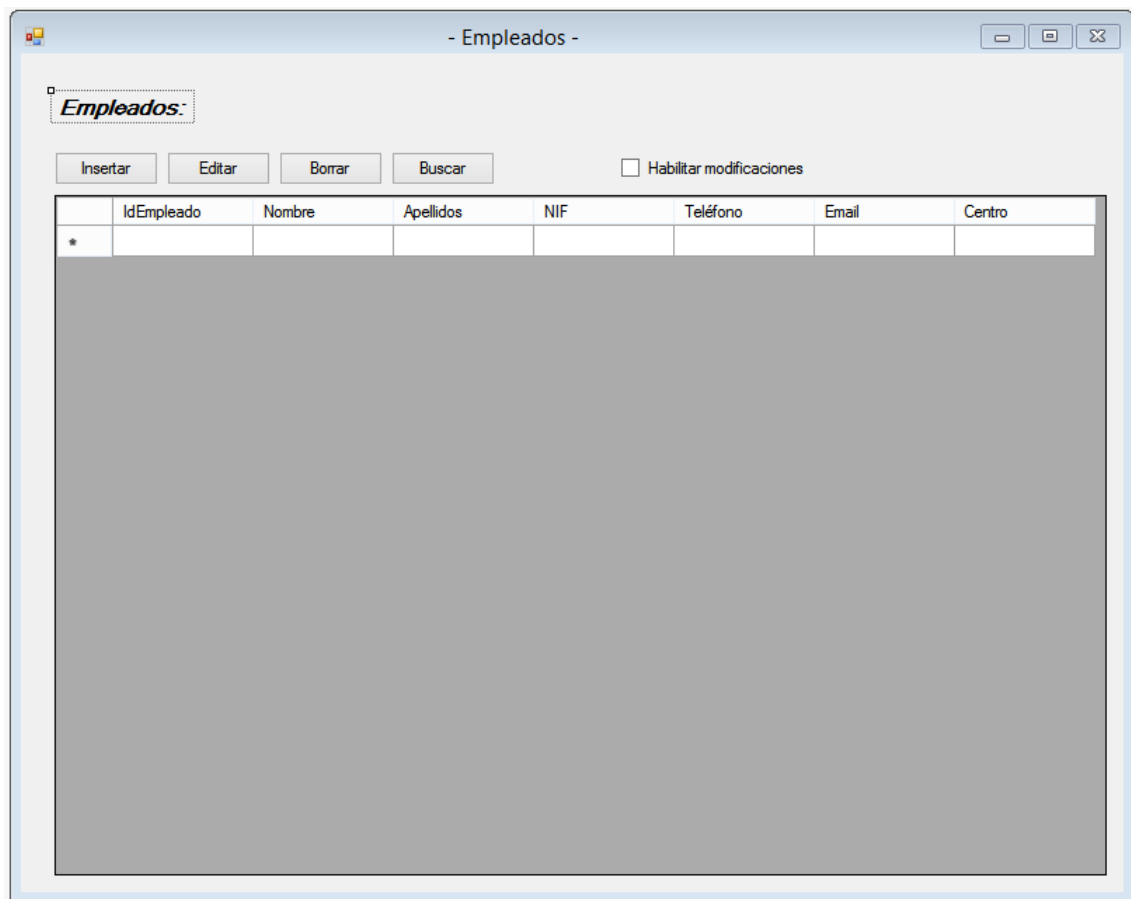


Ilustración 26 - Interfaz de usuario Empleados

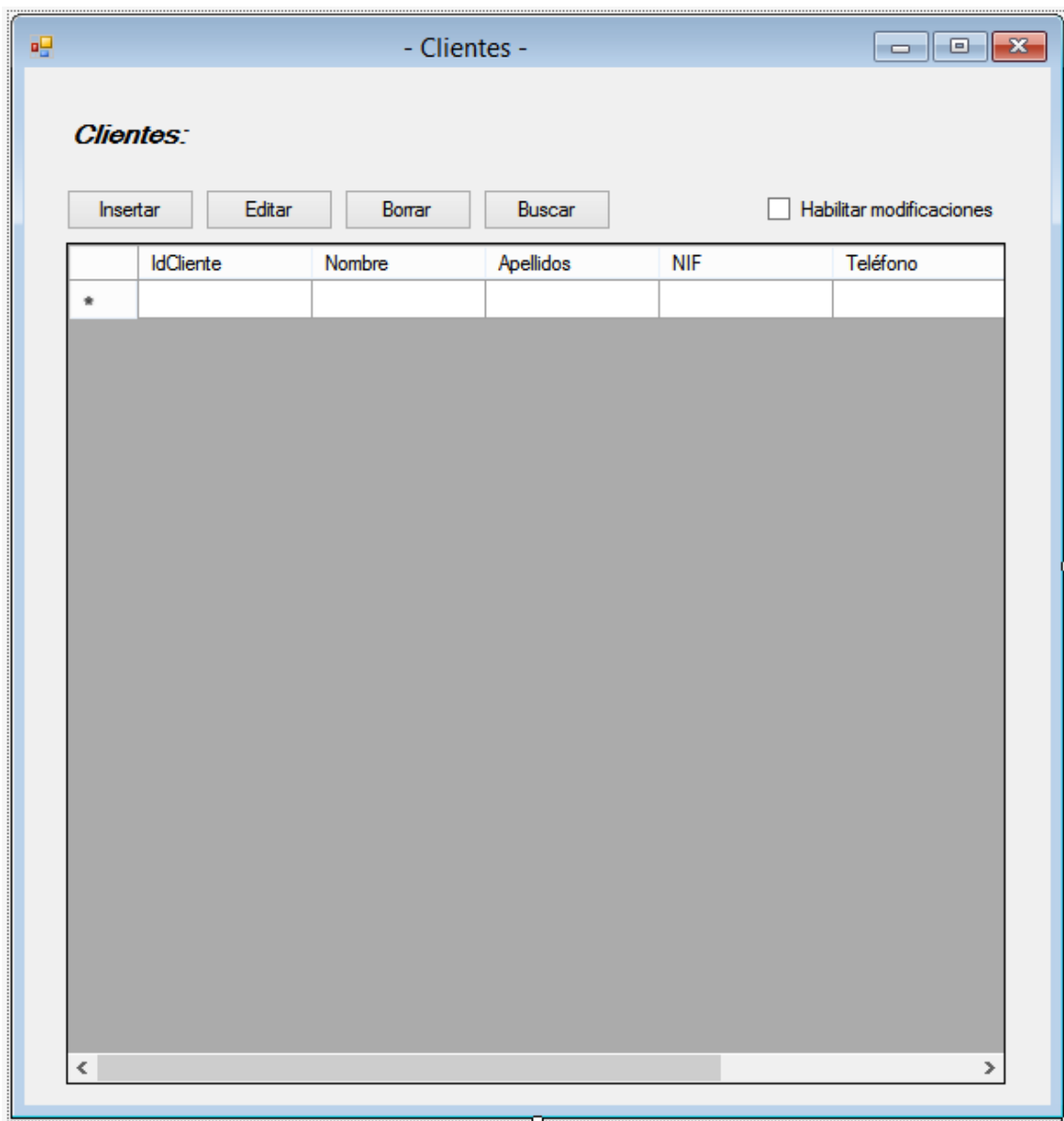


Ilustración 27 - Interfaz de usuario Clientes

Finalmente se expone el estado resultante de las tareas pertenecientes al proyecto:

Tarea	Historia de Usuario	Estado inicial	Estado estimado	Estado final
T01	HU03	Completada	Completada	Completada
T02	HU03	Completada	Completada	Completada
T03	HU03	Completada	Completada	Completada

T04	HU03	Completada	Completada	Completada
T05	HU03	Completada	Completada	Completada
T06	HU03	Completada	Completada	Completada
T07	HU03	Completada	Completada	Completada
T08	HU03	Completada	Completada	Completada
T09	HU03	Completada	Completada	Completada
T10	HU03	Completada	Completada	Completada
T11	HU05	Completada	Completada	Completada
T12	HU05	Completada	Completada	Completada
T13	HU05	Completada	Completada	Completada
T14	HU05	Completada	Completada	Completada
T15	HU05	Completada	Completada	Completada
T16	HU06	Completada	Completada	Completada
T17	HU06	Completada	Completada	Completada
T18	HU06	Completada	Completada	Completada
T19	HU06	Completada	Completada	Completada
T20	HU06	Completada	Completada	Completada
T21	HU06	Completada	Completada	Completada
T22	HU07	No iniciada	No iniciada	No iniciada
T23	HU10	Completada	Completada	Completada
T24	HU10	Completada	Completada	Completada
T25	HU04	Completada	Completada	Completada
T26	HU04	Completada	Completada	Completada
T27	HU04	Completada	Completada	Completada
T28	HU06	Completada	Completada	Completada
T29	HU06	Completada	Completada	Completada
T30	HU01	Completada	Completada	Completada
T31	HU01	Completada	Completada	Completada
T32	HU01	Completada	Completada	Completada
T33	HU02	No iniciada	No iniciada	No iniciada
T34	HU02	No iniciada	No iniciada	No iniciada
T35	HU02	No iniciada	No iniciada	No iniciada
T36	HU02	No iniciada	No iniciada	No iniciada
T37	HU09	No iniciada	No iniciada	No iniciada
T38	HU09	No iniciada	No iniciada	No iniciada
T39	HU09	No iniciada	No iniciada	No iniciada
T40	HU08	Completada	Completada	Completada
T41	HU01	Completada	Completada	Completada
T42	HU01	Completada	Completada	Completada
T43	HU01	Completada	Completada	Completada
T44	HU01	Completada	Completada	Completada
T45	HU01	Completada	Completada	Completada
T46	HU01	Completada	Completada	Completada
T47	HU01	Completada	Completada	Completada

T48	HU01	Completada	Completada	Completada
T49	HU01	Completada	Completada	Completada
T50	HU01	Completada	Completada	Completada

Y el resumen de esfuerzo dedicado a la iteración:

Historia de usuario	Tiempo estimado	Tiempo final
HU01	40	37,5
Total	40	37,5

- Pruebas Funcionales:

Id	P0301
Resultado	Válido
Sistema	Cliente / Base de datos
Descripción	Comprobar que se insertan correctamente registros al realizar una venta.
Acciones	<ol style="list-style-type: none"> 1. Iniciar SGBD 2. Iniciar cliente -> Ventas. 3. Insertar un registro. 4. Realizar consulta de las tablas ventas y stocks.

Id	P0302
Resultado	Válido
Sistema	Cliente / Base de datos
Descripción	Comprobar que asigna correctamente el precio del producto vendido.
Acciones	<ol style="list-style-type: none"> 1. Seleccionar un artículo de la tabla artículos y obtener su precio. 2. Realizar la venta de ese artículo. 3. Consultar la venta realizada y comprar el precio asignado con el precio obtenido de la tabla artículos.

Id	P0303
Resultado	Válido
Sistema	Cliente / Base de datos
Descripción	Comprobar que se puede consultar todas las ventas realizadas anteriormente.
Acciones	1. Realizar una consulta utilizando la interfaz de usuario el botón de buscar y no aplicando ningún parámetro.

Id	P0304
Resultado	Válido
Sistema	Cliente / Base de datos
Descripción	Comprobar la correcta asignación de la fecha al realizar una venta.
Acciones	1. Realizar una venta. 2. Realizar una consulta de la venta realizada y comprobar la correcta asignación de la fecha.

Cuarto Sprint

Historias de usuario restantes:

Historia de Usuario	Miembro			Estimación media	Prioridad
	A	B	C		
HU01					Media
HU02	20	20	20	20	Media
HU03					Alta
HU04					Alta
HU05					Alta

HU06					Alta
HU07	5	3	8	5,33333333	Baja
HU08					Alta
HU09	13	8	13	11,33333333	Baja
HU10					Alta

Se observa que únicamente quedan por desarrollar las historias de usuario 2, 7 y 9.

La estimación de la iteración quedaría de la siguiente forma:

Horas máximas	40
Horas utilizadas	36,66667
Horas restantes	3,333333
Historia	Horas estim.
HU02	20
HU07	5,333333
HU09	11,333333

Por lo que si el esfuerzo estimado es correcto, esta debería ser la última iteración del proyecto, quedando finalizado el desarrollo planificado.

Las historias de usuario a desarrollar son las siguientes:

Historia de Usuario	
ID	HU02
Nombre	Introducir compras realizadas.
Prioridad	Media
Riesgo	Bajo
Descripción	Como gestor de compras quiero introducir las operaciones de compra de artículos realizadas en el sistema.

Validación	<ul style="list-style-type: none"> - Quiero que la compra se guarde en el sistema. - Quiero que la compra guarde una relación de proveedor, empleado que la realiza, el centro de destino y el precio pagado por la compra.
-------------------	---

Historia de Usuario	
ID	HU07
Nombre	Apariencia de la aplicación
Prioridad	Baja
Riesgo	Bajo
Descripción	Como usuario quiero que la aplicación tenga un aspecto simple y sencillo de manejar.
Validación	<ul style="list-style-type: none"> • Quiero que las ventanas sean reducidas y contengan solo la información imprescindible. • Quiero que para entrar en algún modo se abra una ventana específica desde una ventana principal. • Quiero unas funciones básicas (comprar y vender) de fácil acceso.

Historia de Usuario	
ID	HU09
Nombre	Importación / Exportación desde Excel.
Prioridad	Baja
Riesgo	Bajo
Descripción	Como usuario quiero introducir y extraer datos en formato Excel.
Validación	<ul style="list-style-type: none"> • Quiero importar datos desde un fichero .xls. • Quiero exportar datos a un fichero .xls.

La primera historia de usuario que se va a abarcar es la HU02 siguiendo el orden de prioridades.

En esta historia de usuario se pide realizar un desarrollo similar al de la HU01 ya hecho en el anterior sprint, pero en este caso basado en la tabla de compras.

Se crea la tabla compras en la base de datos y se crean sus procedimientos almacenados para la gestión de la tabla.

Después se crea en la capa datos la clase DCompras.cs para interactuar con la tabla de compras de la base de datos.

Se crea en la capa de negocio la clase NCompras.cs y en la capa vista la interfaz de usuario para administrar las compras realizadas.

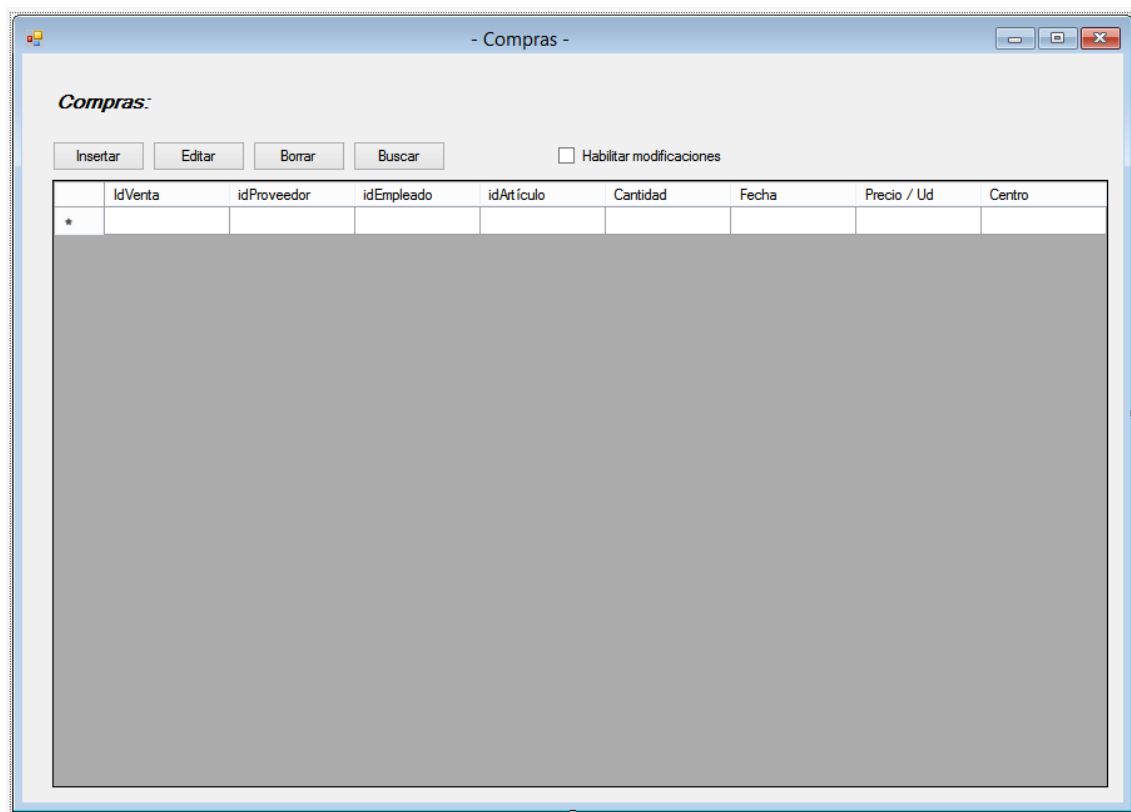


Ilustración 28 - Interfaz de usuario Compras

El siguiente paso será realizar la historia de usuario 7, pero teniendo en cuenta la HU09 a la hora de agregar el botón para la importación de Excel.

En la HU07 se narra la necesidad de disponer de una interfaz de usuario sencilla y que además contenga unos accesos rápidos a determinadas funciones como comprar y vender, en este caso se ha incluido la opción rápida de importar, ya que se ha determinado que es el mejor lugar para situarla.

A parte de las interfaces ya desarrolladas en historias de usuario anteriores, se ha desarrollado la ventana de inicio de la aplicación:

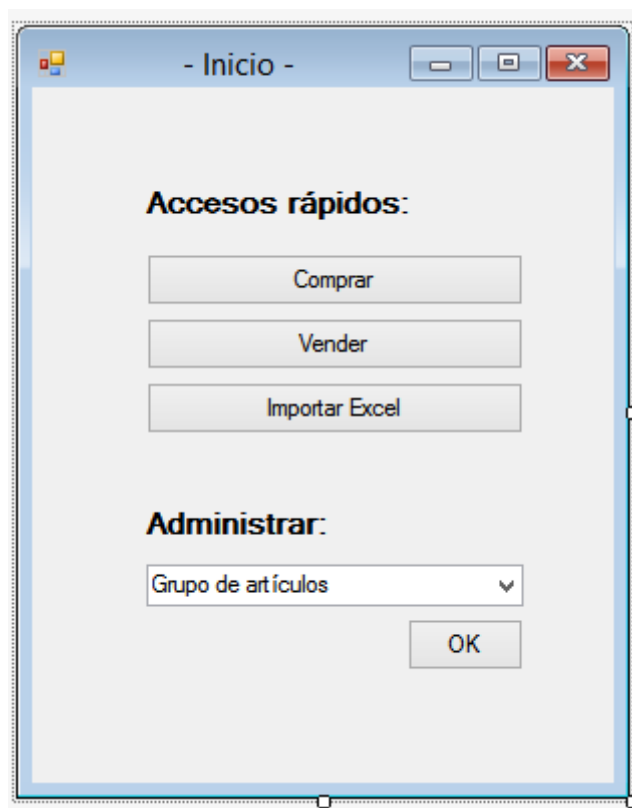
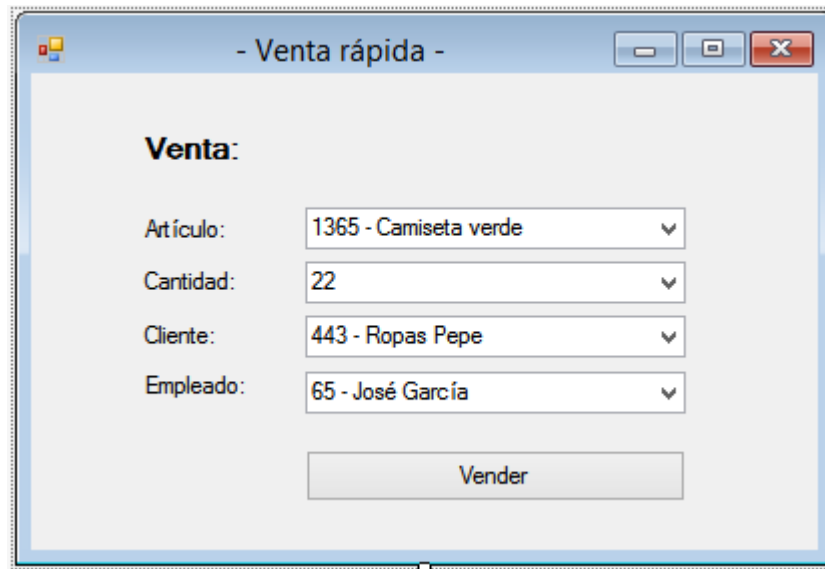


Ilustración 29 - Interfaz de usuario Inicio

En la interfaz de la ventana de inicio se cuenta con tres accesos rápidos a las funciones que se solicitaba de comprar, vender y además añadida de importar desde Excel. Desde cada botón se abre una nueva ventana con la funcionalidad correspondiente. En la parte de administración dispone de un ComboBox en el cual están todas las tablas del sistema. Se selecciona la deseada y se pincha en el botón de Ok para entrar en la ventana de administración de dicha tabla.

Después se han desarrollado las ventanas de compra y venta rápida. Son dos formularios donde el usuario elige entre las opciones del ComboBox, las cuales son extraídas de la base de datos, de modo que solo puede introducir

datos válidos. Para finalizar existe el botón Comprar / Vender con el cual se poduce el insert en la base de datos. Primero llamando a la función de la capa negocio, la cual se encargará de llamar a las funciones necesarias de las distintas tablas de la capa datos (Compra, Venta y Stocks).



- Venta rápida -

Venta:

Artículo: 1365 - Camiseta verde

Cantidad: 22

Cliente: 443 - Ropas Pepe

Empleado: 65 - José García

Vender

Ilustración 30 - Interfaz de usuario Venta rápida



- Compra rápida -

Compra:

Artículo: 915 - Folios A4 x100

Cantidad: 6

Proveedor: 301 - Papelería S.A.

Empleado: 65 - José García

Precio /Ud 13,1 €

Comprar

Ilustración 31 - Interfaz de usuario Compra rápida

Finalmente se aborda la última historia de usuario de la iteración y también del proyecto, la historia de usuario 9. Esta historia de usuario exige implementar una funcionalidad que permita importar y exportar datos desde documentos Excel al sistema desarrollado.

Lo primero que se hará para este desarrollo será agregar una nueva referencia a la capa negocio a objetos del tipo Excel.

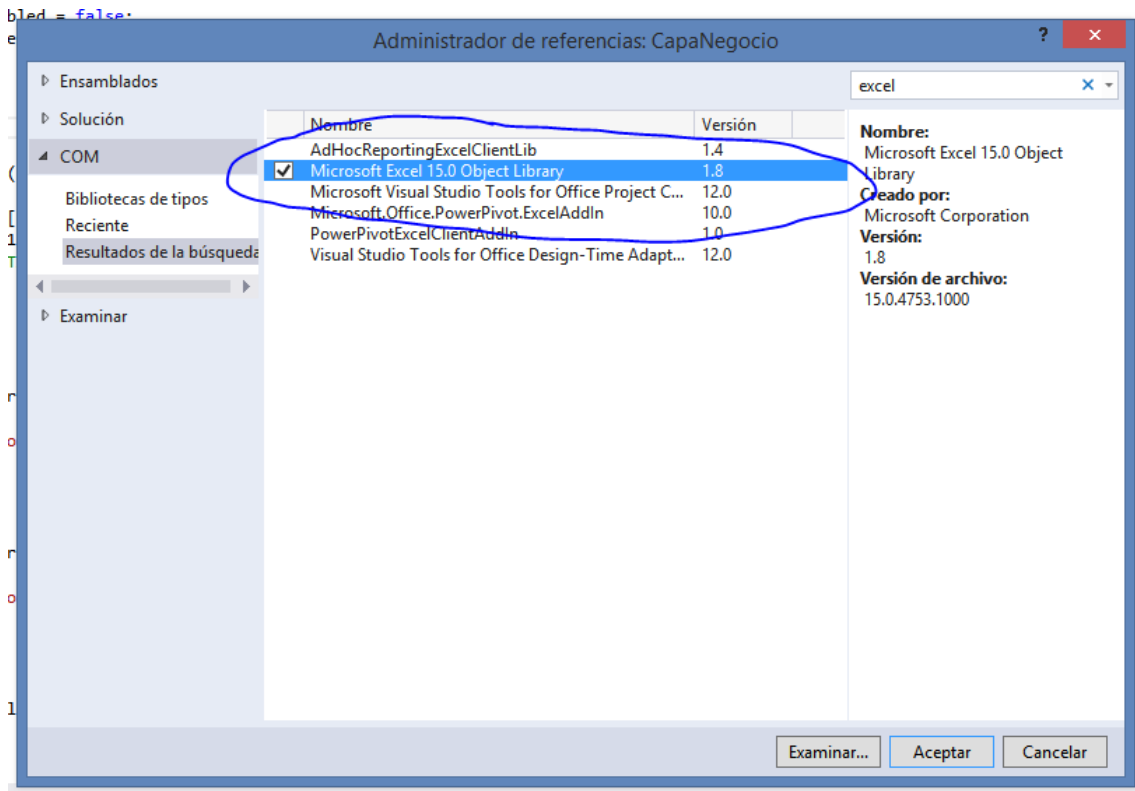


Ilustración 32 - MS Excel Obj. Library

Ahora añadiendo la línea `using Excel = Microsoft.Office.Interop.Excel;` podremos crear objetos del tipo Excel que nos permitirán interactuar con hojas existentes o crear nuevas hojas nuevas.

La interfaz de usuario de esta funcionalidad es la siguiente:

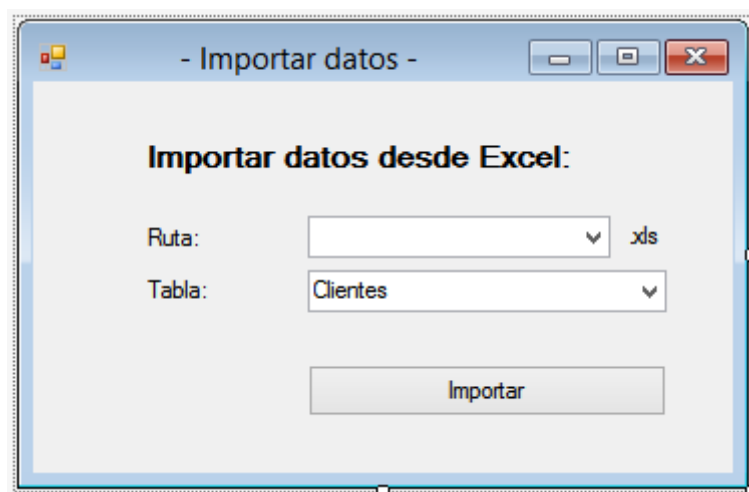


Ilustración 33 - Interfaz de usuario Importar

Con todo esto quedaría finalizado el cuarto sprint y el proyecto al haber completado todas las historias de usuario. A pesar de ello se trata de un proyecto con grandes posibilidades de ampliación de funcionalidades de forma relativamente sencilla y barata, en el siguiente apartado se analizaran estas opciones de futuro.

El resumen de Sprint 4 es el siguiente:

Tarea	Historia de Usuario	Estado inicial	Estado estimado	Estado final
T01	HU03	Completada	Completada	Completada
T02	HU03	Completada	Completada	Completada
T03	HU03	Completada	Completada	Completada
T04	HU03	Completada	Completada	Completada
T05	HU03	Completada	Completada	Completada
T06	HU03	Completada	Completada	Completada
T07	HU03	Completada	Completada	Completada
T08	HU03	Completada	Completada	Completada
T09	HU03	Completada	Completada	Completada
T10	HU03	Completada	Completada	Completada
T11	HU05	Completada	Completada	Completada
T12	HU05	Completada	Completada	Completada
T13	HU05	Completada	Completada	Completada
T14	HU05	Completada	Completada	Completada
T15	HU05	Completada	Completada	Completada
T16	HU06	Completada	Completada	Completada
T17	HU06	Completada	Completada	Completada
T18	HU06	Completada	Completada	Completada

T19	HU06	Completada	Completada	Completada
T20	HU06	Completada	Completada	Completada
T21	HU06	Completada	Completada	Completada
T22	HU07	Completada	Completada	Completada
T23	HU10	Completada	Completada	Completada
T24	HU10	Completada	Completada	Completada
T25	HU04	Completada	Completada	Completada
T26	HU04	Completada	Completada	Completada
T27	HU04	Completada	Completada	Completada
T28	HU06	Completada	Completada	Completada
T29	HU06	Completada	Completada	Completada
T30	HU01	Completada	Completada	Completada
T31	HU01	Completada	Completada	Completada
T32	HU01	Completada	Completada	Completada
T33	HU02	Completada	Completada	Completada
T34	HU02	Completada	Completada	Completada
T35	HU02	Completada	Completada	Completada
T36	HU02	Completada	Completada	Completada
T37	HU09	Completada	Completada	Completada
T38	HU09	Completada	Completada	Completada
T39	HU09	Completada	Completada	Completada
T40	HU08	Completada	Completada	Completada
T41	HU01	Completada	Completada	Completada
T42	HU01	Completada	Completada	Completada
T43	HU01	Completada	Completada	Completada
T44	HU01	Completada	Completada	Completada
T45	HU01	Completada	Completada	Completada
T46	HU01	Completada	Completada	Completada
T47	HU01	Completada	Completada	Completada
T48	HU01	Completada	Completada	Completada
T49	HU01	Completada	Completada	Completada
T50	HU01	Completada	Completada	Completada

Y la tabla de esfuerzo final dedicado a la iteración:

Historia de usuario	Tiempo estimado	Tiempo final
HU02	20	17
HU07	5,333333	5
HU09	11,33333	15

Total	36,66667	37
--------------	----------	----

- Pruebas Funcionales:

Id	P0401
Resultado	Válido
Sistema	Base de datos
Descripción	Mediante el SGBD insertar un registro en la tabla compras.
Acciones	<ol style="list-style-type: none"> 1. Iniciar el SGBD. 2. Introducir un registro en la tabla compras. 3. Realizar una consulta del contenido de la tabla compras.

Id	P0402
Resultado	Válido
Sistema	Cliente / Base de datos
Descripción	Comprobar la correcta asignación de la fecha al realizar una compra.
Acciones	<ol style="list-style-type: none"> 1. Realizar una compra. 2. Realizar una consulta de la compra realizada y comprobar la correcta asignación de la fecha.

Id	P0403
Resultado	Válido
Sistema	Cliente
Descripción	Comprobar que todas las funcionalidades del sistema son accesibles desde la pantalla principal.
Acciones	<ol style="list-style-type: none"> 1. Abrir la aplicación. 2. Entrar en Comprar. 3. Entrar en Vender. 4. Entrar en Importar. 5. Comprobar que en el ComboBox están todas las tablas del sistema.

--	--

Id	P0404
Resultado	Válido
Sistema	Cliente / Base de datos
Descripción	Comprobar que importa los datos desde un Excel de forma correcta.
Acciones	<ol style="list-style-type: none"> 1. Abrir la aplicación. 2. Pinchar en Importar Excel. 3. Poner la ruta del archivo .cls 4. Seleccionar la tabla destino. 5. Pinchar en Importar. 6. Realizar una consulta de la tabla seleccionada y comprobar la correcta inserción de los registros.

Capítulo 5: Planificación y presupuesto

En este capítulo se detallarán los apartados relativos a la planificación y los costes económicos del proyecto.

Planificación

La planificación del desarrollo de cada historia de usuario se muestra en el siguiente diagrama de Gantt

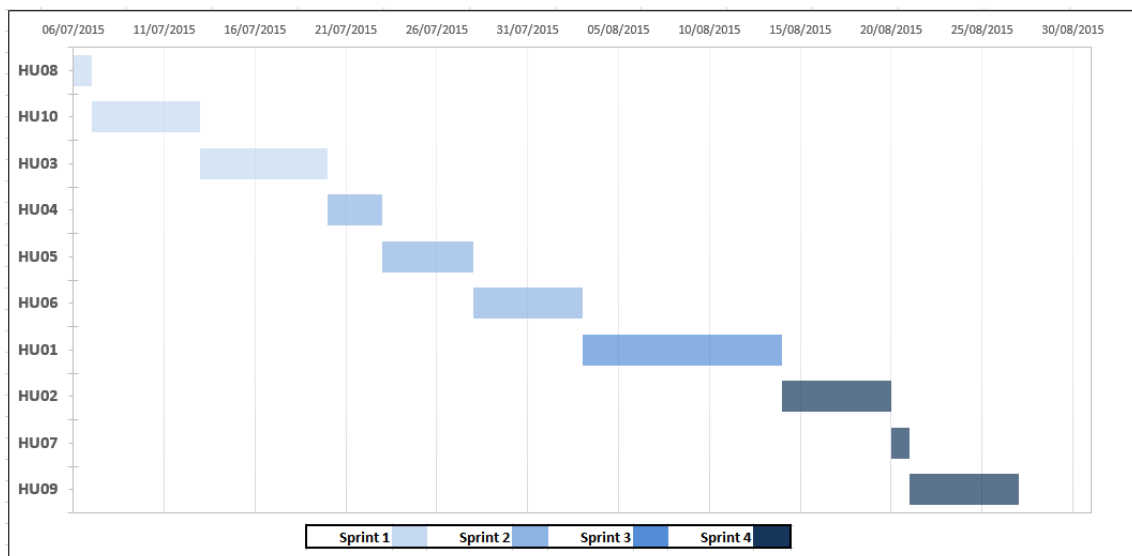


Ilustración 34 - Diagrama de Gantt

Presupuesto

En este apartado se detalla el presupuesto que se estima para llevar a cabo el proyecto.

Se desglosarán los costes en función de su naturaleza. Inicialmente se calcularán todas las cantidades sin aplicar el I.V.A. y finalmente se le añadirá a los resultados totales del proyecto.

Resumen de tiempo dedicado

Según las estimaciones de esfuerzo mostradas a lo largo de este documento se puede extraer el número de horas dedicadas a cada parte del proyecto. Se utilizarán las historias de usuario como referencia para dividir el proyecto en partes del desarrollo, además se le añade una fase adicional correspondiente al análisis y estudio previo y otra fase final correspondiente a la documentación.

Historia de usuario o fase del proyecto	Tiempo estimado (horas)
Estudio previo y análisis	40
HU08	3,666666667
HU10	17,66666667
HU03	17,66666667
HU04	11,33333333
HU05	11,33333333
HU06	15,33333333
HU01	40
HU02	20
HU07	5,333333333
HU09	11,33333333
Documentación	40
Total	233,6666667

Coste de personal

Para calcular los costes relativos al personal se deben tener en cuenta los roles mencionados anteriormente que formarían el equipo y se calculan las horas utilizadas por cada uno.

Empleado	Papel	Dedicación (horas)	Coste (hombre/ hora)	Coste total
Empleado Nº1	Product Owner	40	50,00 €	2.000,00 €
Empleado Nº2	Scrum Master	71	35,00 €	2.485,00 €
Empleado Nº3	Team Member	123	35,00 €	4.305,00 €
			Total	8.790,00 €

Coste de Hardware

Para el desarrollo del sistema es necesario contar con un ordenador para el equipo, por lo que se debe adquirir uno.

Elemento	Coste	% Uso dedicado al proyecto	Dedicación (meses)	Periodo de depreciación (meses)	Coste imputable
Dell Latitude 3550 Laptop	669,00 €	100	2	36	37,17 €

El coste imputable es el resultado de dividir el número de meses que se empleará el equipo entre el número de meses de vida útil que tiene multiplicado por el uso dedicado al proyecto y el coste de adquisición del equipo (sin I.V.A.).

Coste de Software

En este apartado se detalla el software utilizado para el desarrollo del proyecto:

Elemento	Coste
Office Hogar y Empresas 2013	279,00 €
Windows 8.1 Pro	139,00 €
Microsoft Visual Studio Professional 2013	609,70 €
Microsoft SQL Server Developer 2012	612,00 €
Total	1.639,70€

Resumen de costes

Finalmente tras analizar los costes por partes del proyecto se suman todos ellos y se tiene el coste total del proyecto. Después se ha de añadir una previsión de fondos como costes indirectos para cubrir los posibles riesgos del proyecto y gastos imprevistos no indicados en el presupuesto. Estos costes se establecen en un 10% del total.

Concepto	Coste
Personal	8.790,00€
Hardware	37,17€
Software	1.639,70€
Costes indirectos (10%)	1.046,69€
Total sin I.V.A.	11.513,55€
Total con I.V.A.	13.931,40€

Capítulo 6: Conclusiones

En este apartado se va a analizar los resultados obtenidos en el desarrollo del proyecto, así como lo experimentado durante el transcurso de este en relación a la metodología empleada.

Lo primero a detallar son los resultados en cuestión de esfuerzo de dedicado en comparación con el esfuerzo que se ha ido estimando durante el proyecto. A continuación se presentan los datos estimados y los reales:

Sprint	Esfuerzo Estimado (horas)	Esfuerzo Real (horas)	Acumulado Estimado (horas)	Acumulado Real (horas)
0	0	0	0	0
1	39	39,5	39	39,5
2	38	37	77	76,5
3	40	37,5	117	114
4	36,66666	37	153,66666	151

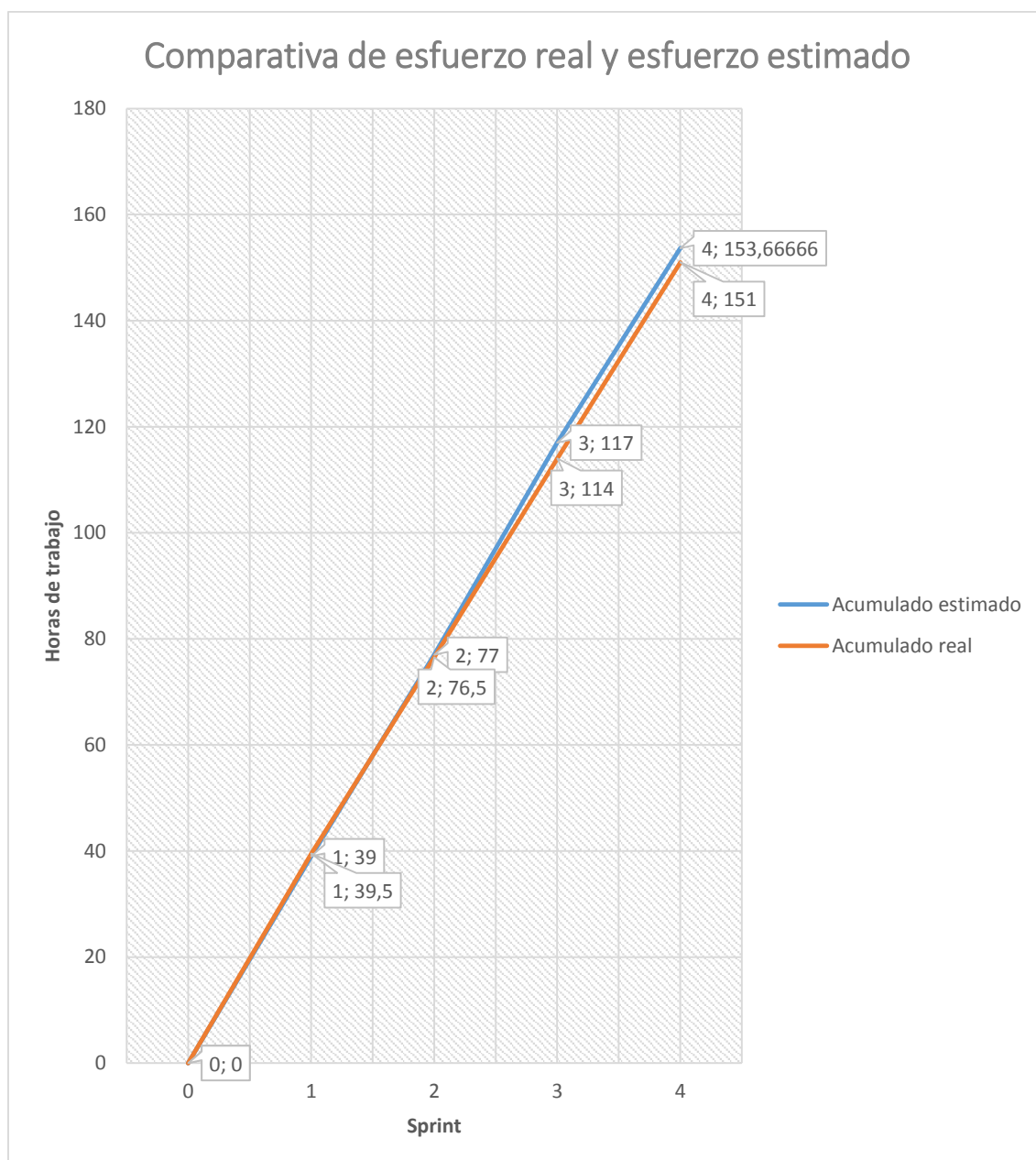


Ilustración 35 - Comparativa esfuerzos

En los gráficos anteriores se puede observar como el rendimiento final ha sido superior al estimado, aunque en general ha sido bastante aproximado. Esto puede deberse a la propiedad adaptativa de la metodología, ya que al permitir reestimaciones en cada iteración, da la posibilidad de utilizar la experiencia actual en el mismo proyecto para afinar la estimación más que si hubiese que quedarse con los valores previos al inicio del proyecto. Especialmente beneficioso en casos en los que la experiencia del equipo en proyectos similares sea baja y por tanto la estimación tenga mayor margen de error.

Respecto a los objetivos iniciales del proyecto se ha cumplido lo que se buscaba en esencia de este trabajo:

- **Utilización de una metodología ágil de desarrollo:** se ha empleado con éxito la metodología Scrum, por lo que se cumple con este punto.
- **Desarrollo de una aplicación utilizable por el usuario medio:** en el diseño de las interfaces se ha buscado que fuese todo lo más simple posible, facilitando al usuario sin experiencia poder utilizar todas las funcionalidades que el sistema le ofrece.
- **Modularidad:** tal como se detallará en el apartado dedicado a trabajos futuros, la aplicación está ideada para recibir actualizaciones con nuevos módulos que implementen nuevas funcionalidades.
- **Persistencia de la información:** se ha incluido una base de datos donde la información se almacena entre sesiones y dispositivos, así todos los registros almacenados en el sistema son persistentes.

Finalmente se puede concluir con que los objetivos marcados inicialmente para el proyecto a desarrollar se han cumplido, por lo que puede considerarse exitoso el proyecto.

Desarrollos futuros

El Proyecto realizado podría decirse que una de sus características fundamentales son las posibilidades de ampliación que tiene, ya que debido a su naturaleza forma una base muy buena para la inclusión de nuevas funcionalidades de manera relativamente sencilla, ya que su estructura está ideada para facilitar la integración de nuevos componentes.

La capacidad de crecimiento en este tipo de aplicación es muy importante, ya que de este modo el sistema puede adaptarse mucho mejor a las necesidades del cliente. De modo que sea la aplicación la que crezca hacia la forma de trabajar de la empresa y no al revés, obligando a esta a realizar importantes y costosas reestructuraciones en sus procesos internos.

La idea de este sistema para la ampliación es poder añadir nuevas verticales de negocio completas aprovechando el núcleo ya existente a modo de datos maestros.

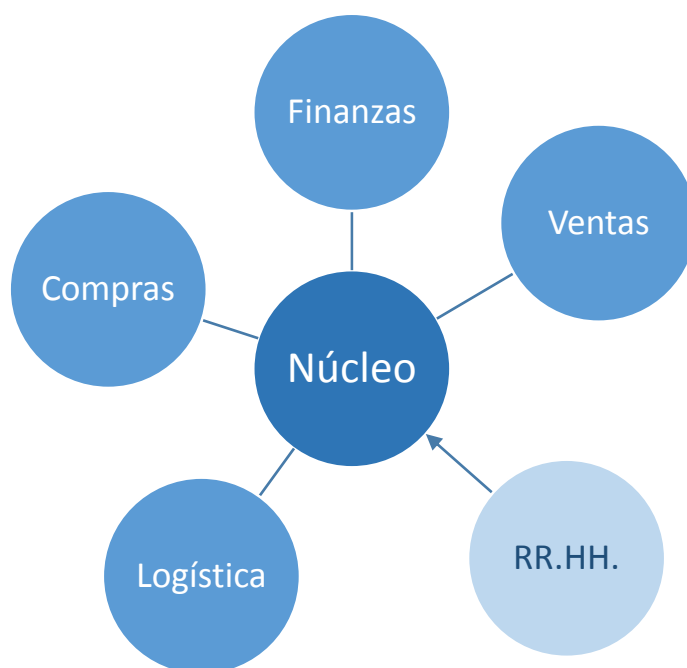


Ilustración 36 - Partes aplicación

Otra facilidad que tiene el sistema para admitir ampliaciones es el diseño de la interfaz de usuario basado en diferentes ventanas. Esto permite añadir funcionalidades sin necesidad de modificar anteriores paneles a excepción de una pantalla de inicio desde donde se podrá acceder al resto de funciones.

Entre estas opciones de futuro se podría destacar la inclusión de funcionalidades como las siguientes:

- **Módulo de gestión financiera:**

Esta ampliación podría dotar de una capacidad al sistema de gestionar las finanzas de la empresa de forma completa y exhaustiva. Así como llevar la contabilidad y analizar los flujos financieros de la empresa. De este modo el departamento financiero puede gestionar de manera más eficiente las cuentas de la empresa, resultando especialmente beneficioso en el caso de que cuente con varios centros.

- **Módulo de gestión de personal:**

En este módulo se ampliaría la posibilidad de gestionar a la plantilla de la empresa, de forma que se incluiría en el sistema toda la información relativa a los datos de los empleados, como vacaciones, salarios, organización, etc.

- **Módulo de gestión de los proveedores:**

Ampliación de la parte de compras de la aplicación. Sirve para optimizar todo el proceso de compras de materiales o servicios a los proveedores. También puede utilizarse para garantizar el control sobre las operaciones de gasto que realizan los empleados añadiendo autorizadores de gasto, presupuestos de compras a los centros, catálogos de productos actualizables por el proveedor, etc.

- **Cliente multiplataforma:**

Actualmente el cliente es únicamente utilizable en un sistema operativo Windows de escritorio, pero una opción a futuro para mejorar la operatividad del sistema sería el desarrollo de versiones web y aplicaciones para Android e iOS. De modo que el usuario pudiese acceder al sistema en cualquier momento y en cualquier lugar, agilizando enormemente los procesos en determinadas circunstancias.

Conclusiones personales

Como experiencia personal, la realización de este Trabajo Fin de Grado ha sido una puesta en escena de los múltiples conocimientos adquiridos durante el resto de la carrera. Especialmente de las asignaturas correspondientes a la especialidad cursada: Sistemas de información.

De esta área de conocimiento estaba especialmente interesado en el campo de la gestión de proyectos así como de las metodologías ágiles de desarrollo, ya que ahora mismo se encuentran en auge y significa un enfoque diferente al tradicional, aportando nuevas formas de hacer las cosas y resultando muy interesante su aplicación a muchos proyectos.

Por otra parte, también existía y existe actualmente un gran interés en el software de gestión de empresas por mi parte. Ya que considero que profesionalmente es un campo en el cual hay buenas posibilidades de negocio, puesto que prácticamente toda empresa dispone de un software de este estilo, ya sea una gran suite en una multinacional o un pequeño programa libre en una empresa de pocos empleados, todo ello forma un buen mercado en el que desarrollarse como profesional. Además la experiencia adquirida en uno de estos sistemas (SAP) por motivos de trabajo ha resultado muy inspirador y útil a la hora de adentrarse en este campo.

La aplicación de la metodología Scrum ha resultado muy gratificante e interesante y me ha servido para conocer esta forma diferente de trabajar. De esta experiencia he sacado la conclusión de que su uso puede resultar muy beneficioso a determinados proyectos, tanto en términos de costes como de ayuda al desarrollador.

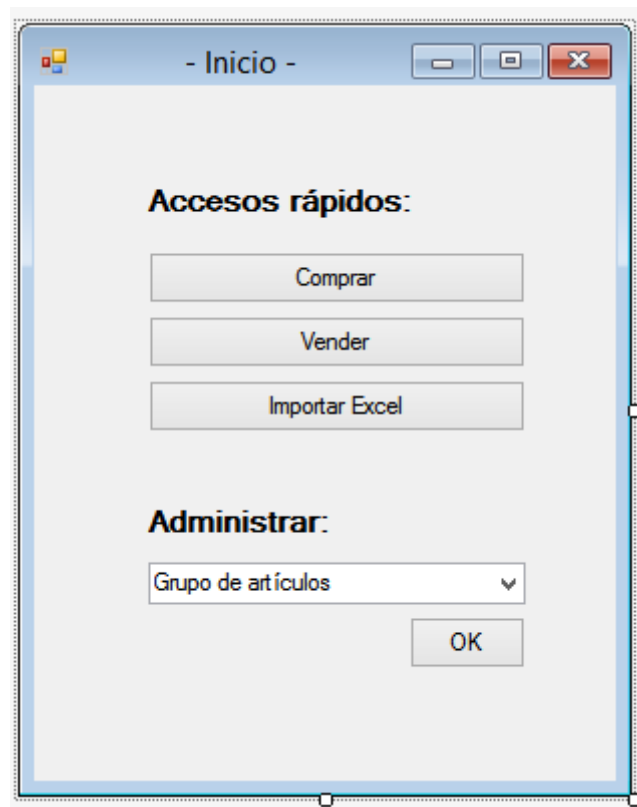
En definitiva, la experiencia de este TFG ha sido muy positiva, me ha permitido acercarme de una forma más global a un proyecto en todas sus fases, a diferencia de las prácticas tradicionales realizadas durante la carrera que están enfocadas a tareas más concretas y relacionadas con la asignatura. De este modo considero que es una forma ideal de finalizar la carrera y empezar un nuevo ciclo en la vida con toda la experiencia adquirida en estos años.

Anexo A: Manual de usuario

En este manual de usuario se explicará el manejo de la aplicación desarrollada, para que un usuario inexperto pueda desenvolverse sin problemas en la aplicación.

1. Inicio de la aplicación.

Nada más iniciar la aplicación aparecerá la siguiente interfaz:



En ella observamos 4 opciones diferentes:

- **Área de accesos rápidos:**
 - *Botón Comprar:* sirve para acceder a la función de compra rápida de la aplicación.

- *Botón Vender:* sirve para acceder a la función de venta rápida de la aplicación.
-
- *Botón Importar Excel:* sirve para acceder a la función de seleccionar una hoja de Excel e insertar sus datos en una tabla del sistema.

- **Área de administración:**

En la pestaña seleccione la tabla en la que desea realizar modificaciones, inserciones o eliminación de algún registro. Con la tabla deseada seleccionada pinche en el botón OK.

2. Compra rápida.

Cuando acceda a la opción de compra rápida desde el menú principal se le abrirá una ventana como la siguiente:


- Para comprar un artículo debe rellenar los siguientes campos:

- *Artículo*: seleccione un artículo de los disponibles en el sistema en la lista desplegable.
- *Cantidad*: introduzca la cantidad deseada o selecciónela en la lista desplegable.
- *Proveedor*: seleccione el proveedor al que comprar el artículo en la lista desplegable.
- *Empleado*: seleccione el empleado responsable de la compra en la lista desplegable.
- *Precio/Ud.*: seleccione el precio por unidad que le indique el proveedor para la compra del producto.

Finalmente, cuando tenga seleccionados todos los datos del formulario pinche en el botón “Comprar”.

3. Venta rápida.

Cuando acceda a la opción de compra rápida desde el menú principal se le abrirá una ventana como la siguiente:



The image shows a software window titled "- Venta rápida -". It contains a form with the following fields:

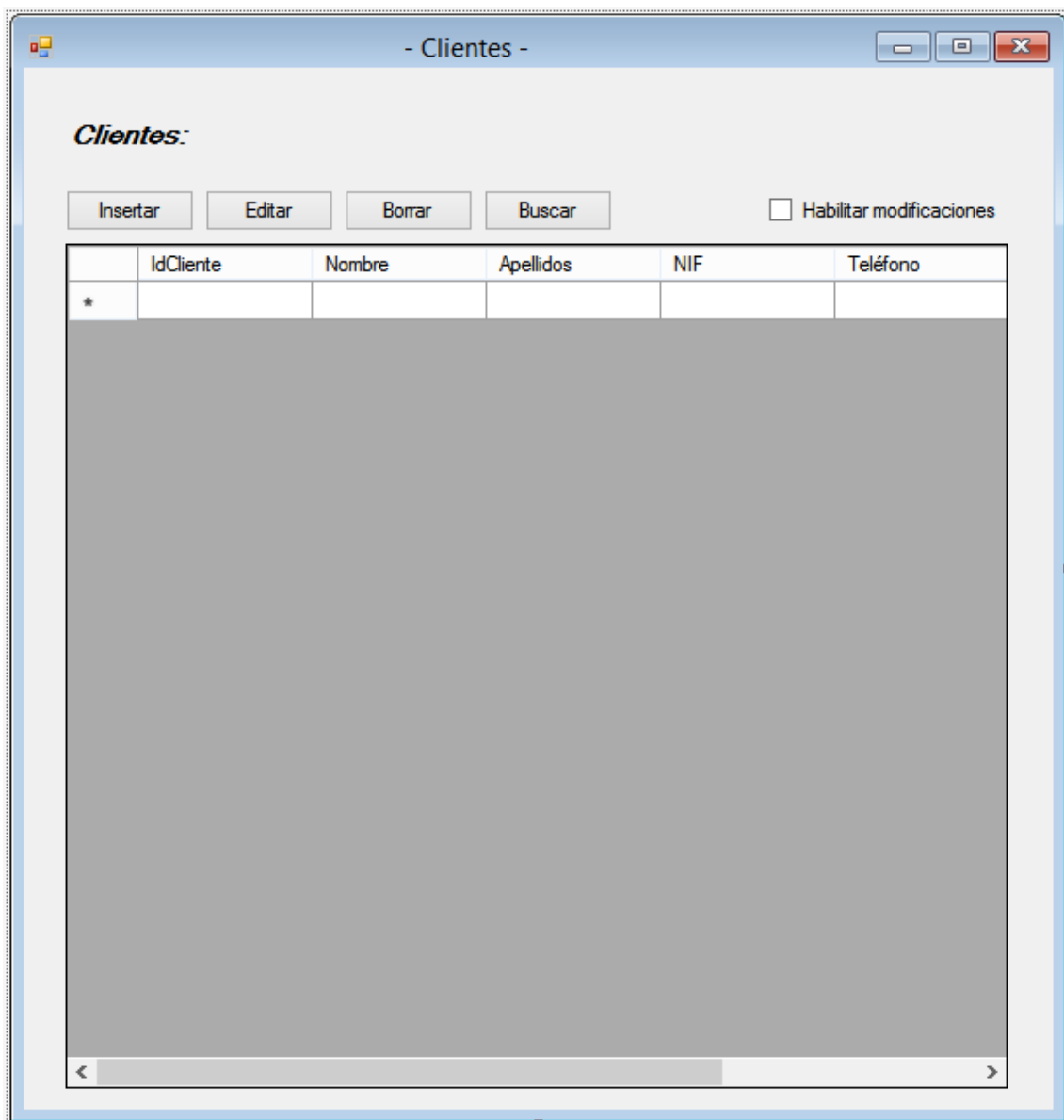
- Venta:**
- Artículo:** 1365 - Camiseta verde
- Cantidad:** 22
- Cliente:** 443 - Ropas Pepe
- Empleado:** 65 - José García
- Vender** (button)

- Para vender un artículo debe rellenar los siguientes campos:
 - Artículo: seleccione un artículo de los disponibles en el sistema en la lista desplegable.
 - Cantidad: introduzca la cantidad deseada o selecciónela en la lista desplegable.
 - Cliente: seleccione el cliente al que vender el artículo en la lista desplegable.
 - Empleado: seleccione el empleado responsable de la venta en la lista desplegable.

Finalmente, cuando tenga seleccionados todos los datos del formulario pinche en el botón “Vender”.

4. Administración de tablas.

Cuando acceda a administrar cualquier tabla del sistema se le abrirá una nueva ventana con la tabla seleccionada. Dentro de esta pantalla dispone de las opciones para la gestión de la tabla.



En la imagen se muestra la ventana de administración de la tabla Clientes, pero estas instrucciones son igualmente válidas para cualquier tabla puesto que tienen el mismo funcionamiento:

- Para que el uso de los botones Editar y Borrar tengan resultado debe estar marcada la opción de Habilitar modificaciones.
- *Botón insertar:* inserta un registro con los datos introducidos en la primera línea de la parte de resultados.

- *Botón Editar:* debe introducir un id que ya exista en el sistema y automáticamente se sustituye el registro antiguo por el nuevo introducido.
- *Botón borrar:* borra el registro seleccionado de la lista de resultados.
- *Botón buscar:* busca el registro o los registros que coincidan con los datos introducidos. Se puede introducir desde todos los datos hasta ninguno. En caso de no introducir ninguno se mostrarán todos los registros de la tabla.

Anexo B: Bibliografía

[1]http://afalco.hol.es/papers/a_modified_agile_methodology_for_an_erp_academic_project_development.pdf

[2]<http://epubs.scu.edu.au/cgi/viewcontent.cgi?article=1274&context=theses>

[3]<http://www.niwotridge.com/PDFs/AgileMethodsforERP.PDF>

[4]http://paper.ijcsns.org/07_book/201108/20110826.pdf

[5]<http://www.forbes.com/sites/louiscolumbus/2013/05/12/2013-erp-market-share-update-sap-solidifies-market-leadership/>

[6]<http://www.evaluandoerp.com/nota-2559-SAP-ORACLE-o-MICROSOFT.html>

[7]<http://www.raima.com/products/product-comparison/>

[8]<https://www.scrum.org/>

[9]www.scrumguides.org/

[10]<https://www.scrumalliance.org/>

[11]K. Molokken - Ostvold y NC Haugen (K Moløkken-Østvold, NC Haugen (10–13 April 2007). "Combining Estimates with Planning Poker—An Empirical Study". 18th Australian Software Engineering Conference (IEEE): 349–58. doi:10.1109/ASWEC.2007.15. Retrieved 1 February 2008.)

[12]<http://blog.iandavis.com/2008/12/what-are-the-benefits-of-mvc/>

[13][https://msdn.microsoft.com/es-es/library/ms190782\(v=SQL.120\).aspx](https://msdn.microsoft.com/es-es/library/ms190782(v=SQL.120).aspx)

[14] <http://blog.xebia.com/2011/02/01/scrum-effective-sprint-zero/>

